

# MPU-6000 和 MPU-6050

## 寄存器映射和说明

### 3.2 版

## 1. 历史版本

略.

## 2. 目的和范围

运动处理单元 Mpu6050 和 Mpu6000 在本文里统称为 MPU-60X0™或者 MPU™。本文主要介绍了这两种传感器关于寄存器映射和说明的基本信息。

Mpu 运动传感器提供了世界上第一个整合了 6 轴运动处理器的解决方法，消除了加速度计和陀螺仪封装级交叉轴失调和离散解。这类处理器把三轴加速度计、三轴陀螺仪和数字运动处理器（DMP）融合在一块芯片上。DMP 运用专业的现场验证运动融合引擎能够处理复杂的 9 轴传感器数据融合算法。MPU-6000 和 MPU-6050 的九轴运动融合算法通过辅助主控 IIC 总线存取外部磁力计或者其他传感器的数据，从而使得设备可以在不介入系统处理器的情况下获得一组完整的数据。这类设备像 MPU-3000™家族一样采用 4x4x0.9 mm QFN 封装，同时进行的简单的路径升级并且实现了在空间本来就受限的电路板更容易布局。为了能精确地追踪快速和慢速这两种运动，MPU 提供了用户可编程的陀螺仪，其满量程变化范围分别为 $\pm 250$ ,  $\pm 500$ ,  $\pm 1000$ , and  $\pm 2000^\circ/\text{sec}$  (dps)。该模块的加速度计同样也可以实现量程用户可编程，其满量程范围分别为 $\pm 2g$ ,  $\pm 4g$ ,  $\pm 8g$ ,  $\pm 16g$ 。

MPU-6000 系列由 MPU-6000 和 MPU-6500 这两类组成。而这两种传感器出了在两方面不相同外，其他方面完全一致。MPU-6050 支持通讯频率高达 400kHz 的 IIC 通讯方式，并且有一个 VLOGIC 引脚来定义接口电压水平。MPU-6000 除了支持 IIC 通讯方式外，还支持高达 20MHz 的 SPI 通讯方式，并且只有一个供电引脚 VDD，它同时作为该模块的逻辑参考电平和模拟电平。

为了获取关于 MPU-60X0 更详细的信息，请参考文献“MPU-6000 and MPU-6050 Product Specification”。

## 3. 寄存器映射

MPU-60X0 的寄存器映射列出如下。

表格略.

注意：寄存器名字以\_H 和\_L 结尾的寄存器分别表示一个内部数据的高字节和低字节。

在这份详细的寄存表中，寄存器名称用大写字母表示，寄存器的值用大写字母和斜体表示。比如寄存器 `ACCEL_XOUT_H`（第 59 号），其值表示为 *ACCEL\_XOUT\_H*（15:8），用来表示加速度计 X 轴 16 位测量值的值的高八位。

除了下面的两个寄存器，其他寄存器的初始化值均为 0X00：

- 1. Register 107: 0x40;
- 2. Register 117: 0x68。

## 4. 寄存器说明

这个部分描述了 MPU-60X0 内部的所有寄存器的功能和内容。设备在上电时从睡眠模式开启。

### 4.1 寄存器1-辅助IIC供电选择：AUX\_VDDIO。

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
01	1	AUX_VDDIO	-	-	-	-	-	-	-

说明：这个寄存器指定了辅助 IIC 的供电电压水平。

MPU-6050: *AUX\_VDDIO* 用来设置辅助 IIC 的高逻辑电平是来自 VDD 还是 VLOGIC.

MPU-6000: *AUX\_VDDIO* 应该被置为 0。

位 0-6 被保留。

参数：

*AUX\_VDDIO*                      MPU-6050: 被置位的时候，辅助 IIC 的高逻辑电平来自 VDD；被清零的时候高逻辑电平来自 VLOGIC。

MPU-6000: 置零。

### 4.2 寄存器25-采样率分频器：SMPRT\_DIV

Type: Read/Write

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

这个寄存器指定陀螺仪输出频率的分频值以作为 MPU-6050 的采样率。寄存器输出、FIFO 输出、DMP 采样、运动检测、零点运动检测、自由落体检测的频率都是以采样频率为基础。采样率是通过陀螺仪的输出频率通过分频得到的，分频数方式通过 *SMPLRT\_DIV* 确定：

采样率=陀螺仪输出率/(1+*SMPLRT\_DIV*)

当 DLPF 被禁用的时候 (*DLPF\_CFG* = 0 or 7)，陀螺仪输出频率=8kHz；当 DLPF 使能的时候，陀螺仪采样率为 1kHz。（见寄存器 26）。

**说明：**加速度计的输出频率为 1kHz，也就是说，当采样率大于 1kHz 的时候，同一个加速度计采样值可能会不止一次送入 FIFO、DMP 以及传感器寄存器中。

为了图示加速度计和陀螺仪的信号路径，请参照“MPU-6000/MPU-6050 Product Specification document”的 Section 8。

参数：

*SMPLRT\_DIV* 为 8 位无符号数。采样率就是利用该值通过分频陀螺仪输出率得到的。

4.3 寄存器26-设置配置：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]		

说明：

这个寄存器用来配置加速度计和陀螺仪的外部帧同步（external Frame Synchronization 【FSYNC】）引脚采样以及设置数字低通滤波器（Digital Low Pass Filter 【DLPF】）。

一个连接到 FSYNC 接口的外部信号可以通过配置 *EXT\_SYNC\_SET* 被采样。FSYNC 引脚上的信号变化被锁存，这样很短的脉冲信号也可以被捕获。被锁存的 FSYNC 信号通过采样频率被采样，而采样率通过寄存器 25 进行配置。当锁存器中的数据被采样之后，锁存器中的值改变为当前 FSYNC 引脚的信号。

通过采样得到的值通过传送，取代某一传感器数据寄存器的最低有效位。而该寄存器的类型通过 *EXT\_SYNC\_SET* 来确定，具体的对应表格如下表所示：

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]

DLPF（数字低通滤波器）是通过 *DLPF\_CFG* 配置的。加速度计和陀螺仪的值的滤波方式由 *DLPF\_CFG* 确定，对应的表格如下：

DLPF_CFG	Accelerometer (F <sub>s</sub> = 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

参数：

- EXT\_SYNC\_SET*3 位无符号数，定义 FSYNC 引脚采样；
- DLPF\_CFG*3 位无符号数，配置 DLPF。

4.4 寄存器27-陀螺仪配置：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]		-	-	-

这个寄存器用来触发陀螺仪的自检以及设定陀螺仪的满测量范围。陀螺仪自检机制允许用户测试其自身的机械性能和电子性能。陀螺仪每个轴的自检可以通过控制该寄存器的 *XG\_ST*、*YG\_ST* 和 *ZG\_ST*

位来触发。每个轴的自检可以同步或者独立进行。

当自检开始的时候，板上的电子器件会刺激相应的传感器。这种刺激会使传感器的质量块（proof masses）移动一段适当的距离，这段距离可以抵消已经定义了的 coriolis force（科里奥利力，地球自转偏向力）。质量块的移位导致了传感器输出的变化，这种变化在输出信号中体现出来。输出信号用来观察自检响应。

**自检响应定义如下：**

自检响应=自检使能的传感器输出-自检禁止的传感器输出。

陀螺仪每个轴的自检的正常允许范围在《MPU-6000/MPU-6050 Product Specification document》中的电气特性表可以看到。当某一轴的自检响应在最大/最小允许范围之内，那么这部分的自检就通过了，反之则认为自检没有通过。

*FS\_SEL* 决定了陀螺仪的满量程输出范围，对应的表格如下所示：

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

位 2-0 保留。

**参数：**

- XG\_ST*     置位 X 轴自检；
- YG\_ST*     置位 Y 轴自检；
- ZG\_ST*     置位 Z 轴自检。
- FS\_SEL*     两位无符号数，用于选定陀螺仪的满量程范围。

4.5 寄存器28-加速度计配置：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]		

**描述：**

这个寄存器用来触发加速度计的自检并设定其满量程范围。这个寄存器也用来配置数字高通滤波器（Digital High Pass Filter ,DHPF）。

加速度计自检允许用户检测其自身的机械和电气部分性能。加计每个轴的自检可以分别通过控制位  $XA\_ST$ 、 $YA\_ST$  和  $ZA\_ST$  触发，且各个轴的自检可以独立进行，也可以同时进行。

当自检触发的时候，板上电子器件会刺激响应的传感器。这种刺激相当于模拟了一个外力。被刺激的传感器反过来会产生一个对应的输出信号，这个输出信号就会用来观察自检响应。

**自检响应定义如下：**

自检响应=自检使能传感器输出-自检禁止传感器输出。

加速度计每个轴的自检响应范围在《MPU-6000/MPU-6050 Product Specification document》的电气特性表中可以看到。当某一轴的自检响应在最大/最小允许范围之内，那么这部分的自检就通过了，反之则认为自检没有通过。

$AFS\_SEL$  用于选择加速度计的满量程范围，其对应表格如下：

AFS_SEL	Full Scale Range
0	$\pm 2g$
1	$\pm 4g$
2	$\pm 8g$
3	$\pm 16g$

$ACCEL\_HPF$  用来配置数字高通滤波器 DHPF，以使其在运动检测（自由落体，运动阈值，零点漂移：(Free Fall,Motion threshold, and Zero Motion)）中可用。高通滤波器的输出对于数字寄存器是不可用的，见《MPU-6000/MPU-6050 Product Specification document》Section 8 的插图。

高通滤波器有三种模式：

**RESET：**滤波器的输出在一个采样周内的下降至 0，实际上是禁止了高通滤波器，这个模式可以用来快速降低滤波器输出。

**ON：**滤波器使高于界限频率的采样信号通过。

**HOLD：**被触发后，滤波器保持当前值，滤波器的输出为输出样本和保持样本之差。

ACCEL_HP	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

**参数：**

$XA\_ST$  置位加计 X 轴自测；

- YA\_ST      置位加计 Y 轴自测；
- ZA\_ST      置位加计 Z 轴自测。
- ACCEL\_FS\_SEL      两位无符号数，选择加速度计的满量程范围。
- ACCEL\_HPF      三位无符号数，配置高通滤波器。

4.6 寄存器29-自由落体加速度阈：FF\_THR

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1D	29	FF_THR[7:0]							

描述：

这个寄存器为自由落体时间探测设置探测阈值。*FF\_THR* 的每一位所表示的重力增量可以在《MPU-6000/MPU-6050 Product Specification document》中的电气特性表中找到。

当加速度计的每个轴所测得的加速度的绝对值均比这个探测阈值小的话，启动自由落体检测。这个条件也将启动自由落体时间计数器（寄存器 30）。当自由落体计数器的计数值达到在 *FF\_DUR* (Register 30)中设定的值时，自由落体中断被触发。

关于自由落体中断更详细的信息，请查看《MPU-6000/MPU-6050 Product Specification document》Section 8.2 以及本文中的寄存器 56 和 58。

参数：

*FF\_THR*      8 位无符号数，指定自由落体探测阈值。

4.7 寄存器 30-自由落体时间：FF\_DUR

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1E	30	FF_DUR[7:0]							

描述：



该寄存器为自由落体探测事件的时间计数器设置阈值。时间计数器的计数频率为 1kHz，也就是说 FF\_DUR 的 1LSB=1ms。

当加速度计的每个轴所测得的加速度的绝对值均比这个探测阈值小的话，启动自由落体检测。这个条件也将启动自由落体时间计数器（寄存器 30）。当自由落体计数器的计数值达到在 FF\_DUR (Register 30)中设定的值时，自由落体中断被触发。

于自由落体中断更详细的信息，请查看《MPU-6000/MPU-6050 Product Specification document》Section 8.2 以及本文中的寄存器 56 和 58。

**参数:**

FF\_DUR 八位无符号数， 指定时间计数器阈值。1LSB = 1ms。

**说明:** 本寄存器在用到的时候回给一个初值作为计数器阈值，但同时又会以 1kHz 的速度增加，故而应该是计数溢出触发自由落体中断。

4.8 寄存器 31-运动检测阈值：MOT\_THR

**Type: Read/Write**

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1F	31	MOT_THR[7:0]							

**描述:**

这个寄存器为运动中断的产生配置探测阈值。*MOT\_THR* 的每一位所表示的重力增量可以在《MPU-6000/MPU-6050 Product Specification document》中的电气特性表中找到。

当加速度计三轴的测量值的绝对值都比运动阈值大的时候，运动被检测。这个条件满足则运动检测时间计数器（寄存器 32）开始增加计数。当时间计数器的值达到 *MOT\_DUR* (Register 32)的设定值的时候，触发运动检测中断。

运动中断会在 *MOT\_DETECT\_STATUS* (Register 97)指出对应的轴的极性。

关于运动中断更详细的信息，请查看《MPU-6000/MPU-6050 Product Specification document》Section 8.3 以及本文中的寄存器 56 和 58。

**参数:**

**MOT\_THR** 八位无符号数，指定运动检测阈值。

4.9 寄存器 32-运动检测时间：MOT\_DUR

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
20	32	MOT_DUR[7:0]							

描述：

这个寄存器为运动中断的产生配置时间计数器阈值。时间计数器的计数频率为 1kHz，也就是说 MOT\_DUR 的 1LSB=1ms。

当加速度计三轴的测量值的绝对值都比检测阈值大的时候，运动被检测。这个条件满足则运动检测时间计数器（寄存器 32）开始增加计数。当时间计数器的值达到 MOT\_DUR (Register 32)的设定值的时候，触发运动检测中断。

关于运动中断更详细的信息，请查看《MPU-6000/MPU-6050 Product Specification document》Section 8.3 以及本文中的寄存器 56 和 58。

参数：

MOT\_DUR 八位无符号数， 指定时间计数器阈值， 1 LSB = 1ms。

4.10 寄存器 33-静态检测（Zero Motion）检测阈值：ZRMOT\_THR

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
21	33	ZRMOT_THR[7:0]							

描述：

这个寄存器为静态检测中断的产生配置时间计数器阈值。ZRMOT\_THR 的每一位所表示的重力增量可以在《MPU-6000/MPU-6050 Product Specification document》中的电气特性表中找到。

当加速度计三轴的测量值的绝对值都比检测阈值小的时候，静态被检测。这个条件满足则静态检测

时间计数器（寄存器 34）开始增加计数。当时间计数器的值达到 *ZRMOT\_DUR* (Register 34)的设定值的时候，触发静态检测中断。

跟自由落体和运动检测不同的是，零运动中断在第一次零检测或不再进行零检测的时候触发中断。  
当静止状态事件被检测到时，*MOT\_DETECT\_STATUS* 寄存器 (Register 97)中的静止状态标志位讲会被置位或清零。当满足运动-静止时，该标志位置位；当满足静止-运动时，该标志位被清零。

关于静态检测中断更详细的信息，请查看《MPU-6000/MPU-6050 Product Specification document》Section 8.4 以及本文中的寄存器 56 和 58。

**参数：**  
*ZRMOT\_THR*      八位无符号数，指定静态运动检测阈值。

**4.11 寄存器 34-静态检测时间：ZRMOT\_DUR**

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
22	34	ZRMOT_DUR[7:0]							

**描述：**  
这个寄存器为静态运动中断的产生配置时间计数器。时间计数器的技术频率为 16Hz，因此 *ZRMOT\_DUR* 的最低位表示的时间为 64ms，即 1 LSB = 64 ms。

当加速度计三轴的测量值的绝对值都比检测阈值(Register 33)小的时候，静态被检测。当时间计数器的值达到 *ZRMOT\_DUR* (Register 34)的设定值的时候，触发静态检测中断。

关于静态检测中断更详细的信息，请查看《MPU-6000/MPU-6050 Product Specification document》Section 8.4 以及本文中的寄存器 56 和 58。

**参数：**  
*ZRMOT\_DUR*      八位无符号数，指定时间计数器阈值，1 LSB = 64ms。

**4.12 寄存器 35-FIFO 使能：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

#### 描述:

这个寄存器的内容决定了哪一个传感器的测量值被送到 FIFO 中。

如果某一传感器的 FIFO\_EN 位被置位，那么存储在该传感器数据寄存器(Registers 59 to 96)中的数据将会被传送到 FIFO 中。

当某传感器的 FIFO\_EN 位被置位，传感器数据寄存器中的数据会被传送到 FIFO 中。这个传感器以在寄存器 25 中设置的采样率被采样。关于数据寄存器的更多信息，请参考寄存器 59-96。

当一个外部从机对应的 FIFO\_EN (*SLVx\_FIFO\_EN*, where x=0, 1, or 2)被置位，存储在相应数据寄存器(EXT\_SENS\_DATA 寄存器, 寄存器 73 -96)中的数据就会以采样率的速度写入 FIFO。

EXT\_SENS\_DATA 寄存器和 I2C 从设备的联系是靠 I2C\_SLVx\_CTRL 寄存器 (where x=0, 1, or 2; 寄存器 39,42, 45)决定的。关于寄存器 EXT\_SENS\_DATA 的信息，请参考寄存器 73-96。

注意，从设备 3 的 FIFO\_EN 位(*SLV3\_FIFO\_EN*)在寄存器 I2C\_MST\_CTRL (Register36)中。另外需要注意的是，从设备 4 与从设备 0-3 的工作方式不相同。关于从设备 4，请参考寄存器 49-53 以获取更多信息。

#### 参数:

*TEMP\_FIFO\_EN* 被置位后，TEMP\_OUT\_H 和 TEMP\_OUT\_L (寄存器 65 and 66)中的数据可以写入到 FIFO 中。

*XG\_FIFO\_EN* 被置位后，GYRO\_XOUT\_H 和 GYRO\_XOUT\_L(Registers 67 and 68) 中的数据可以被写入到 FIFO 中。

*YG\_FIFO\_EN* 被置位后，GYRO\_YOUT\_H 和 GYRO\_YOUT\_L(Registers 69 and 70) 中的数据可以被写入到 FIFO 中。

*ZG\_FIFO\_EN* 被置位后，YRO\_ZOUT\_H 和 GYRO\_ZOUT\_L(Registers 71 and 72) 中的数据可以被写入到 FIFO 中。

*ACCEL\_FIFO\_EN* 被置位后，ACCEL\_XOUT\_H, ACCEL\_XOUT\_L, ACCEL\_YOUT\_H, ACCEL\_YOUT\_L, ACCEL\_ZOUT\_H,和 ACCEL\_ZOUT\_L (寄存器 59 to 64) 中的数据可以被写入到 FIFO 中。

*SLV2\_FIFO\_EN* 被置位后，和从设备 2 联系在一起的数据寄存器 EXT\_SENS\_DATA ( 寄存器 73 -96) 中的数据被允许输入到 FIFO 中。

**SLV1\_FIFO\_EN** 被置位后，和从设备 1 联系在一起的数据寄存器 EXT\_SENS\_DATA（寄存器 73-96）中的数据被允许输入到 FIFO 中。

**SLV0\_FIFO\_EN** 被置位后，和从设备 0 联系在一起的数据寄存器 EXT\_SENS\_DATA（寄存器 73-96）中的数据被允许输入到 FIFO 中。

**注意：**关于数据寄存器 EXT\_SENS\_DATA 的更多信息，请参照寄存器 73-96。

### 4.13 寄存器 36-IIC 主机控制：I2C\_MST\_CTRL

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
24	36	MULT_MST_EN	WAIT_FOR_ES	SLV_3_FIFO_EN	I2C_MST_P_NSR	I2C_MST_CLK[3:0]			

描述：

这个寄存器为多主机或单主机配置辅助 IIC 总线。此外，本寄存器被用来延迟数据准备中断（Data Ready interrupt），同时使能从设备 3 的数据写入 FIFO 中。可以配置 MPU 的 8MHz 的内部时钟，可以配置辅助 IIC 主机从读某一从设备转换到读另一个从设备。

多主机容纳能力使得多 IIC 主机可以在同一总线上操作。在需要多主机容纳能力的电路上，配置 **MULT\_MST\_EN**。这将增加大约 30uA 的电流损耗。

在需要多主机容纳能力的回路中，IIC 总线的状态总是被各个独立的 IIC 主机监控着。在一个 IIC 主机控制（仲裁，arbitration）总线之前，它必须首先确定没有其他的 IIC 主机在控制这条总线。当

当 **WAIT\_FOR\_ES** 被置位后，数据准备中断讲被延迟一直到外部从设备的数据被送到 EXT\_SENS\_DATA 数据寄存器中。这个功能是为了保证当数据准备中断被触发的时候内部传感器的数据和外部传感器的数据都被懂到了它们各自对应的数据传感器中。

当从设备 3 的使能位 **SLV\_3\_FIFO\_EN** 被置位后，从设备 3 的传感器测量数据将会被送到 FIFO 中。和 IIC 从设备关联的数据寄存器 EXT\_SENS\_DATA 由寄存器 I2C\_SLV3\_CTRL (Register 48) 决定。

关于寄存器 EXT\_SENS\_DATA 的更多信息，请参考寄存器 73-96。从设备 0, 1, 2 对应的 FIFO\_EN 位可以在寄存器 35 中找到。

I2C\_MST\_P\_NSR 位配置 IIC 主机从读取一个主机转换到读取另一个主机。如果该位为 0，在两次读取之间会有一个重启。如果该位置位，在下一次读取之前会有一个停止。如果一个写事件跟在一个读事件之后，那么下一次读取开始之前将总会用到一次停止。

I2C\_MST\_CLK 是一个无符号四位数，用来为 MPU 内部 8MHz 的时钟配置分频倍数。它根据下表设置 IIC 主机时钟速度：

I2C_MST_CLK	I <sup>2</sup> C Master Clock Speed	8MHz Clock Divider
0	348 kHz	23
1	333 kHz	24
2	320 kHz	25
3	308 kHz	26
4	296 kHz	27
5	286 kHz	28
6	276 kHz	29
7	267 kHz	30
8	258 kHz	31
9	500 kHz	16
10	471 kHz	17
11	444 kHz	18
12	421 kHz	19
13	400 kHz	20
14	381 kHz	21
15	364 kHz	22

**参数：**

- MUL\_MST\_EN

当置位时，多主机控制使能。
- WAIT\_FOR\_ES

置位时，延迟数据准备中断直到从设备外部传感器数据被送到 EXT\_SENS\_DATA 中。
- SLV3\_FIFO\_EN

置位时，与从设备 3 关联的 EXT\_SENS\_DATA 数据寄存器中的数据被送到 FIFO 中，从设备 0-2 对应的功能位可以在寄存器 35 中找到。
- I2C\_MST\_P\_NSR

控制 IIC 主机从读某一从设备到读另一从设备。（from one slave read to the next slave read.）。

当该位等于 0，在两次读设备之间会有一个重启。

当该位等于 1，在下一次读取设备之前会有一个停止和开始标志。

当在读设备之后又一个写操作，则两次操作之间将总会有一个停止和开始标志。
- I2C\_MST\_CLK

四位无符号数。配置 IIC 主机时钟速度的分频倍数。

**注意：**关于数据寄存器 EXT\_SENS\_DATA 的更多信息，请参照寄存器 73-96。

**4.14 寄存器 37-39---IIC 从设备 0 控制：I2C\_SLV0\_ADDR, I2C\_SLV0\_REG, I2C\_SLV0\_CTRL**



### Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
25	37	I2C_SLV0_RW	I2C_SLV0_ADDR[6:0]						
26	38	I2C_SLV0_REG[7:0]							
27	39	I2C_SLV0_EN	I2C_SLV0_BYTE_SW	I2C_SLV0_REG_DIS	I2C_SLV0_GRP	I2C_SLV0_LEN[3:0]			

### 描述:

这些寄存器为从设备 0 设置数据传递序列。从设备 1, 2, 3 跟 0 差不多。但是, 从设备 4 的特性跟 0-3 相差很多。关于从设备 4 的信息, 请参考寄存器 49-53。

MPU60X0 和从机 0 的数据传输通过设置 *I2C\_SLV0\_RW* 位实现读或者写。置位时, 读操作; 清零时, 写操作。

*I2C\_SLV0\_ADDR* 用来定义从设备 0 的 I2C 地址。

数据传输从从设备 0 的一个内部寄存器开始。这个寄存器的地址由 *I2C\_SLV0\_REG* 指定。

传输数据的字节数由 *I2C\_SLV0\_LEN* 指定。当传输的字节数大于 1 时 (*I2C\_SLV0\_LEN* > 1), 对从 *I2C\_SLV0\_REG* 开始的连续地址内的数据进行读 (写)。

在读模式下, 读取的数据被放在可用的 *EXT\_SENS\_DATA* 的最底端的那个里面。关于读取数据的放置问题的更多信息, 请参考 *EXT\_SENS\_DATA* 寄存器 (73-96)。

在写模式下, *I2C\_SLV0\_DO* (Register 99) 的内容将会被写入从设备。

*I2C\_SLV0\_EN* 使能从设备 0 的 I2C 数据传输。只有当多于 0 字节的数据要被传输, 即 (*I2C\_SLV0\_LEN* > 0) 时, 数据传输操作才会被执行, 且此时 *I2C\_SLV0\_EN* = 1。

*I2C\_SLV0\_BYTE\_SW* 配置字符对的字节交换。当字节交换使能, 一个字符对的高字节和低字节被交换。关于字符对的配对规则, 请参照 *I2C\_SLV0\_GRP*。当该位清零时, 写入或者读出从设备 0 的数据时, 这些数据会按照传输顺序先存放在 *EXT\_SENS\_DATA* 数据寄存器中以等待写入或读出。

当 *I2C\_SLV0\_REG\_DIS* 置位时, 只进行数据的读或写操作。当该位清零时, 在读写数据之前会先传输数据所在寄存器地址。当指定从设备中要进行读写操作的寄存器时, 该位应被设为 0。

*I2C\_SLV0\_GRP* 指定从寄存器取出的字符对的组合顺序。当该位清零时, 寄存器 0 和 1, 2 和 3, 等等(偶地址+奇地址)中的数据被组合在一起; 当该位置位时的组合顺序为 (奇地址+偶地址)。

I2C 数据传输速度跟寄存器 25 中设置的采样率一致。用户必须保证在一个采样周期内应该完成对使能从设备的对或写操作。

I2C 从机访问速度可以依据采样率成比例降低。降低的程度由 *I2C\_MST\_DLY* (Register 52) 决定。一

个从设备的访问速度是否可以在采样率的基础上成比降低由 *I2C\_MST\_DELAY\_CTRL*(Register 103)决定。

从设备的处理顺序是固定的。处理从设备的序列为从设备 0、从设备 1、从设备 2、从设备 3、和从设备 4。如果某一个从设备未使能，跳过。

每一个从设备都可以以采样率或比采样率小的速度被访问。当一部分设备的访问率为采样率，而另一些数据的访问速度较采样率小，访问从设备的顺序（从设备 0-4）仍然不变。但是，当访问速度慢的设备的访问率决定在某一次操作从设备的循环中不应该被访问，该设备将会被跳过。关于低访问速率的更多信息，请参考寄存器 52。。无论从机的访问速率是采样频率还是低频率都由寄存器 103 中的 *Delay Enable* 位所决定。

参数：

<i>I2C_SLV0_RW</i>	置位时，读操作；清零时，数据传输为写操作。
<i>I2C_SLV0_ADDR</i>	从设备 0 的 I2C7 位地址。
<i>I2C_SLV0_REG</i>	从设备 0 的 8 位寄存器地址，也是数据传输的起始地址。
<i>I2C_SLV0_EN</i>	被置位时，允许从设备 0 的数据传递操作。被清零时，禁止从设备与主机进行数据传递。
<i>I2C_SLV0_BYTE_SW</i>	置位时，使能字节交换。当字节交换使能时，一个字符组的高字节和低字节互换。关于字符组的组合规则请参考 <i>I2C_SLV0_GRP</i> 。 被清零时，从设备 0 的读写数据按顺序被写入 <i>EXT_SENS_DATA</i> 中。
<i>I2C_SLV0_REG_DIS</i>	置位时，只会进行数据的读操作或者写操作。 清零时，在读或者写数据之前会传输要写或读的寄存器的地址。
<i>I2C_SLV0_GRP</i>	指定来自寄存器的字符组的组合顺序。当该位清零时，寄存器 0 和 1, 2 和 3, 等等(偶地址+奇地址)中的数据被组合在一起；当该位置位时的组合顺序为（奇地址+偶地址）。
<i>I2C_SLV0_LEN</i>	4 位无符号数。指定与从设备 0 要传递数据的字节数。 该位清零表示禁用该寄存器，这与向 <i>I2C_SLV0_EN</i> 写 0 等价。

字节交换示例：

下面的例子展示了当 *I2C\_SLV0\_BYTE\_SW* =1 的字节交换。

*I2C\_SLV0\_GRP* = 0, *I2C\_SLV0\_REG* = 0x01, *I2C\_SLV0\_LEN* = 0x4:

1. 从从设备 0 的寄存器 0x01 中读取的第一个字节数据将被存放在 *EXT\_SENS\_DATA\_00*。因为 *I2C\_SLV0\_GRP* =0，先偶后奇两个地址中的数据被组成字符对。因为本次读操作从一个奇地址



开始的而不是偶地址，故而只有一个字节被读取。

2. 因为  $I2C\_SLV0\_BYTE\_SW = 1$  且  $I2C\_SLV0\_REG[0] = 1$ ，读取到的第二个和第三个字节将互换。  
从 0x02 读取的数据被放在 EXT\_SENS\_DATA\_02 中，从 0x03 中读取的数据会被存放在 EXT\_SENS\_DATA\_01 中。
3. 最后一个读取的字节，也就是从地址 0x04 中读取的字节将会被存储在 EXT\_SENS\_DATA\_03 中。因为在读操作中只有一个字节，所以字节交换没有出现。

#### 从设备访问示例：

从设备 0 的访问速率为采样率，从设备 1 的访问率为采样率的一半，其他的从设备被禁用。在第一个循环中，从设备 0 和从设备 1 均被访问。但是，在第二个循环中，只有从设备 0 被访问。在第三个循环中，从设备 0 和从设备 1 均将会被访问。

### 4.15 寄存器 40-42---I2C 从设备 1 控制：I2C\_SLV1\_ADDR, I2C\_SLV1\_REG, I2C\_SLV1\_CTRL

#### Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
28	40	I2C_SLV1_RW	I2C_SLV1_ADDR[6:0]						
29	41	I2C_SLV1_REG[7:0]							
2A	42	I2C_SLV1_EN	I2C_SLV1_BYTE_SW	I2C_SLV1_REG_DIS	I2C_SLV1_GRP	I2C_SLV1_LEN[3:0]			

描述：

这些寄存器描述了从设备 1 的数据传输序列。他们的功能和描述从设备 0 的寄存器一致。

### 4.16 寄存器 43-45---I2C 从设备 2 控制：I2C\_SLV2\_ADDR, I2C\_SLV2\_REG, I2C\_SLV2\_CTRL

**Type: Read/Write**

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
2B	43	I2C_SLV2_RW	I2C_SLV2_ADDR[6:0]						
2C	44	I2C_SLV2_REG[7:0]							
2D	45	I2C_SLV2_EN	I2C_SLV2_BYTE_SW	I2C_SLV2_REG_DIS	I2C_SLV2_GRP	I2C_SLV2_LEN[3:0]			

描述:

这些寄存器描述了从设备 2 的数据传输序列。他们的功能和描述从设备 0 的寄存器一致。

#### 4.17 从设备 46-48---I2C 从设备 3 控制: I2C SLV3 ADDR, I2C SLV3 REG, I2C SLV3 CTRL

**Type:** Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
2E	46	I2C_SLV3_RW	I2C_SLV3_ADDR[6:0]						
2F	47	I2C_SLV3_REG[7:0]							
30	48	I2C_SLV3_EN	I2C_SLV3_BYTE_SW	I2C_SLV3_REG_DIS	I2C_SLV3_GRP	I2C_SLV3_LEN[3:0]			

描述:

这些寄存器描述了从设备 3 的数据传输序列。他们的功能和描述从设备 0 的寄存器一致。

#### 4.18 寄存器 49-53---I2C 从设备 4 控制: I2C\_SLV4\_ADDR, I2C\_SLV4\_REG, I2C\_SLV4\_DO, I2C\_SLV4\_CTRL, I2C\_SLV4\_DI

## Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
31	49	I2C_SLV4_RW	I2C_SLV4_ADDR[6:0]						
32	50	I2C_SLV4_REG[7:0]							
33	51	I2C_SLV4_DO[7:0]							
34	52	I2C_SLV4_EN	I2C_SLV4_INT_EN	I2C_SLV4_REG_DIS	I2C_MST_DLY[4:0]				
35	53	I2C_SLV4_DI[7:0]							

## 描述：

这些寄存器描述了从设备的数据传递。从设备 4 的特性和从设备 0-3 有很大的不同。关于从设备 0-3 的更多信息，请参考寄存器 37-48。

MPU 和从设备 4 之间通过 I2C 进行的数据传输是通过设置 *I2C\_SLV4\_RW* 位来控制读写的。当该位置位时，读操作；当该位清零时，写操作。

*I2C\_SLV4\_ADDR* 用来定义从设备 4 的 I2C 从地址，为 7 位。

数据传输开始于从设备 4 的一个内部寄存器。这个寄存器的地址定义为 *I2C\_SLV4\_REG*。

在读模式下，读取的结果将被放在 *I2C\_SLV4\_DI* 中。在写模式下，*I2C\_SLV4\_DO* 中的内容将被写入从设备中。

当 *I2C\_SLV4\_EN* 位置位时，数据传输被执行。当 *I2C\_SLV4\_ADDR* 和 *I2C\_SLV4\_REG* 位设置好以后，应立即置位 *I2C\_SLV4\_EN* 位。写操作时，*I2C\_SLV4\_DO* 被用到。当数据传输完成时，*I2C\_SLV4\_EN* 将会被清零。

如果中断使能，从设备 4 完成数据传输后将会触发中断。中断的状态可以在寄存器 54 中看到。

当 *I2C\_SLV4\_REG\_DIS* 位被置位时，数据传输形式为读/写数据而不是写寄存器地址。如果要指定接下来和那个从设备的内部寄存器进行数据传输，该位应清理以指定对应寄存器的地址。

*I2C\_MST\_DLY* 用来设置从设备的访问速度，其是采样率经过分频得到，公式为：

$$1 / (1 + I2C\_MST\_DLY) \text{ samples}$$

采样率是由 *SMPLRT\_DIV* (register 25) 和 *DLPF\_CFG*(register 26)决定的。一个从设备的访问率是否可以相对于采样率降低，是由 *I2C\_MST\_DELAY\_CTRL* (register 103)决定的。

关于采样率的更多信息，请参考寄存器 25。

从设备 4 的数据传输是在从设备 0,1,2,3 的传输完成之后再进行的。因此，从设备 4 的最大访问速度是由寄存器 25 中设置的采样率决定的。

## 参数：

*I2C\_SLV4\_RW*        当被置位时，读操作；当清零时，写操作。

*I2C\_SLV4\_ADDR*    从设备 4 的 7 位 I2C 地址

*I2C\_SLV4\_REG*     从设备 4 的内部寄存器 8 位地址，数据传输从这里开始。

*I2C\_SLV4\_DO*       将要写入从设备 4 的数据存放在这里。如果 *I2C\_SLV4\_RW* 位被置位(读操作)，该寄存器不起作用。

*I2C\_SLV4\_EN*       被置位时，使能从设备 4 数据传输操作。

清零时，禁止从设备 4 的数据传输。

*I2C\_SLV4\_INT\_EN* 置位时，从设备 4 数据传输完成时可以产生一个中断信号。

清零时，从设备 4 数据传输完成时不产生中断信号。

中断的状态可以在寄存器 54 中看。

*I2C\_SLV4\_REG\_DIS* 置位时，传输为读/写数据。

清零时，传输为读/写一个寄存器地址。

*I2C\_MST\_DLY* 设置从设备相对于采样率减小之后的访问速度。

*I2C\_SLV4\_DI* 这个寄存器保存从从设备 4 中读取的数据。在读操作结束后该区域被填充。

#### 4.19 寄存器 54-I2C 主机状态: *I2C\_MST\_STATUS*

**Type: Read Only**

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
36	54	PASS_THROUGH	I2C_SLV4_DONE	I2C_LOST_ARB	I2C_SLV4_NACK	I2C_SLV3_NACK	I2C_SLV2_NACK	I2C_SLV1_NACK	I2C_SLV0_NACK

说明:

这个寄存器描述了 MPU 内部的 IIC 主机产生的中断信号的状态。同时也向主处理器传递 FSYNC 中断状态。

**参数:**

*PASS\_THROUGH* 这个位影响从外部设备进入到 MPU 内部的 FSYNC 中断的状态。被用来通过 MPU 向主应用程序处理器传递一个外部中断。

置位时，如果 *INT\_PIN\_CFG*(Register 55)中的 *FSYNC\_INT\_EN* 被声明，该位将产生一个中断。

*I2C\_SLV4\_DONE* 当从设备 4 数据传输完成后，自动置位。如果 *INT\_ENABLE* (Register 56) 中 *I2C\_MST\_INT\_EN* 被声明且 *I2C\_SLV4\_CTRL* (Register 52)中的 *SLV\_4\_DONE\_INT* 被声明，置位后将触发中断。

*I2C\_LOST\_ARB* 当 I2C 主机失去了对辅助 IIC 的仲裁（发生错误）之后，该位被自动置位。如果寄存器 *INT\_ENABLE* (Register 56)中的 *I2C\_MST\_INT\_EN* 位被声明，则将触发中断。

<i>I2C_SLV4_NACK</i>	当主机收到一个来自从设备 4 的 NACK 信号之后，该位自动置位。如果寄存器 INT_ENABLE (Register 56)中的 I2C_MST_INT_EN 位被声明，则将触发中断。
<i>I2C_SLV3_NACK</i>	当主机收到一个来自从设备 3 的 NACK 信号之后，该位自动置位。如果寄存器 INT_ENABLE (Register 56)中的 I2C_MST_INT_EN 位被声明，则将触发中断。
<i>I2C_SLV2_NACK</i>	当主机收到一个来自从设备 2 的 NACK 信号之后，该位自动置位。如果寄存器 INT_ENABLE (Register 56)中的 I2C_MST_INT_EN 位被声明，则将触发中断。
<i>I2C_SLV1_NACK</i>	当主机收到一个来自从设备 1 的 NACK 信号之后，该位自动置位。如果寄存器 INT_ENABLE (Register 56)中的 I2C_MST_INT_EN 位被声明，则将触发中断。
<i>I2C_SLV0_NACK</i>	当主机收到一个来自从设备 0 的 NACK 信号之后，该位自动置位。如果寄存器 INT_ENABLE (Register 56)中的 I2C_MST_INT_EN 位被声明，则将触发中断。

#### 4.20 寄存器 55-中断引脚/旁路使能配置：INT\_PIN\_CFG

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
37	55	INT_LEVEL	INT_OPEN	LATCH_INT_EN	INT_RD_CLEAR	FSYNC_INT_LEVEL	FSYNC_INT_EN	I2C_BYPASS_EN	CLKOUT_EN

描述：

这个寄存器配置 INT 引脚上中断信号的表现形式。这个寄存器也用来使能 FSYNC 引脚作为主应用程序处理器的一个中断，同时使能 I2C 主机的旁路模式。也有使能时钟输出的功能。

*FSYNC\_INT\_EN* 使能 FSYNC 引脚来作为主应用程序处理器的一个中断源。*FSYNC\_INT\_LEVEL* 中的到有效电平的过渡将会触发一个中断。这个中断的状态在 I2C 主机状态寄存器(Register 54)中的 *PASS\_THROUGH* 位中可以看到。

当 *I2C\_BYPASS\_EN* 为 1 且 *I2C\_MST\_EN* (Register 106 bit[5])等于 0，主应用程序处理器将能够直接

访问 MPU 的辅助 I2C 总线。当该位为 0，不管 *I2C\_MST\_EN* 是什么，主应用程序处理器将不能直接访问 MPU 的 I2C 辅助总线。

关于旁路模式更多的信息，请参考《MPU-6000/MPU-6050 Product Specification document》的 Section7.11 和 Section7.13。

参数：

<i>INT_LEVEL</i>	置位时，INT 引脚的逻辑电平为高电平有效 清零时，INT 引脚的逻辑电平为低电平有效。
<i>INT_OPEN</i>	清零时，INT 引脚配置为推挽式； 置位时，INT 引脚配置为开漏（open drain）即高阻态，在外接上拉电阻的时候，可以实现高电平和低电平的输出。
<i>LATCH_INT_EN</i>	清零时，INT 引脚产生一个 50us 长的脉冲。 置位时，在中断清除之前一直保持高电平。
<i>INT_RD_CLEAR</i>	清零时，中断状态位只有在读取寄存器 INT_STATUS (Register 58)时才会被清除。 置位时，中断状态位在任何读取操作之后均会清除。
<i>FSYNC_INT_LEVEL</i>	清零时，FSYNC 引脚（当作为主处理器的一个中断源时），高电平有效 置位时，FSYNC 引脚（当作为主处理器的一个中断源时），低电平有效。
<i>FSYNC_INT_EN</i>	清零时，禁止 FSYNC 引脚作为主处理器的中断源； 置位时，使能 FSYNC 引脚作为主处理器的中断源。

4.21 寄存器 56-中断使能：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	-	-	DATA_RDY_EN

描述：

这个寄存器通过中断源使能中断的产生。

关于自由落体检测、运动检测、已经静态检测，请参考寄存器 29-34。

关于每个中断源的中断状态的更多信息，请参考寄存器 58。关于 I2C 主机中断产生的跟多信息请参考寄存器 54。

位 1、2 被保留。

参数：

- FF\_EN* 置位时，使能自由落体检测中断的产生。
- MOT\_EN* 置位时，使能运动检测中断的产生。
- ZMOT\_EN* 置位时，使能静态检测中断的产生。
- FIFO\_OFLOW\_EN* 置位时，使能 FIFO 溢出中断的产生。
- I2C\_MST\_INT\_EN* 置位时，使能任何 I2C 主机中断源。
- DATA\_RDY\_EN* 置位时，使能数据准备中断，该中断在完成一次向所有寄存器写操作时产生。

4.22 寄存器 58-中断状态：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	-	-	DATA_RDY_INT

描述：

这个寄存器展示了每个中断源的中断状态。读取该寄存器之后，每一位都将被清除。

关于对应的中断使能位的更多信息，请参考寄存器 56。

位 1、2 为保留位。

- FF\_INT* 自由落体中断产生时，该位自动置位。  
该寄存器被读出后，该位自动清零。
- MOT\_INT* 当运动检测中断产生时自动置位。该寄存器被读出后，该位自动清零。



<i>ZMOT_INT</i>	当静态检测中断产生时自动置位。该寄存器被读出后，该位自动清零。
<i>FIFO_OFLOW_INT</i>	当 FIFO 溢出中断产生时自动置位。该寄存器被读出后，该位自动清零。
<i>I2C_MST_INT</i>	当 I2C 主机中断产生时该位自动置位。I2C 主机中断列表请参考寄存器 54。 该寄存器被读出后，该位自动清零。
<i>DATA_RDY_INT</i>	当数据就绪中断产生时该位自动置位。该寄存器被读出后，该位自动清零。

**4.23 寄存器 59to64-加速度计测量值：ACCEL\_XOUT\_H, ACCEL\_XOUT\_L, ACCEL\_YOUT\_H, ACCEL\_YOUT\_L, ACCEL\_ZOUT\_H, ACCEL\_ZOUT\_L**

**Type: Read Only**

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
3B	59	ACCEL_XOUT[15:8]							
3C	60	ACCEL_XOUT[7:0]							
3D	61	ACCEL_YOUT[15:8]							
3E	62	ACCEL_YOUT[7:0]							
3F	63	ACCEL_ZOUT[15:8]							
40	64	ACCEL_ZOUT[7:0]							

**描述：**

这些寄存器保存加速度计的最近测量值。

加速度计的测量值以在寄存器 25 中设置的采样率写入这些寄存器。

加速度计测量值寄存器、温度寄存器、陀螺仪测量值寄存器、外部传感器数据寄存器都是由两套寄存器组成：一套内部寄存器和一套用户面向的读只寄存器。

加速度计的内部寄存器组中的数据总是以采样率的速度在更新。同时，用户面向的只读寄存器组在串口可用时复制内部寄存器组的数据。这保证了对寄存器组突发读取（burst read，应该是连续地址的字节读取）时可以读取到同一采样率时刻的测量值。注意，如果突发读取没有用到，用户应该通过利用数据就绪中断保证在单字节读取时可以读取到当前采样时刻的值。

16 位加速度计的测量值的满量程范围可以用 *ACCEL\_FS* (Register 28)来定义。对应的灵敏度如下表格所示：



AFS_SEL	Full Scale Range	LSB Sensitivity
0	$\pm 2g$	16384 LSB/mg
1	$\pm 4g$	8192 LSB/mg
2	$\pm 8g$	4096 LSB/mg
3	$\pm 16g$	2048 LSB/mg

参数：

*ACCEL\_XOUT*                    16 位补码值，用来存储 X 轴最近的测量值。

*ACCEL\_YOUT*                    16 位补码值，用来存储 Y 轴最近的测量值。

*ACCEL\_ZOUT*                    16 位补码值，用来存储 Z 轴最近的测量值。

#### 4.24 寄存器 65 和 66-温度测量：TEMP\_OUT\_H and TEMP\_OUT\_L

Type: Read Only

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
41	65	TEMP_OUT[15:8]							
42	66	TEMP_OUT[7:0]							

描述：

这些寄存器存储最近的温度传感器的测量值。温度测量值以采样率写入这些寄存器。

这些温度测量寄存器，以及加速度测量值寄存器、陀螺仪测量值寄存器、外部传感器数据寄存器，都是由两套寄存器组成：一套内部寄存器和一套面向用户的只读寄存器。

传感器内部寄存器内的数据组总是以采样率的速度更新。同时，如果串口可用，那么面向用户的寄存器组复制内部寄存器组的值。这保证了对寄存器组突发读取（burst read，应该是连续地址的字节读取）时可以读取到同一采样率时刻的测量值。注意，如果突发读取没有用到，用户应该通过利用数据就绪中断保证在单字节读取时可以读取到当前采样时刻的值。

量程决定因素已经传感器温度补偿可以在电气规格表中可以看到(《MPU-6000/MPU-6050 Product Specification document》的 Section6.4).

参数：

*TEMP\_OUT*                    16 位有符号数。存储最近的温度传感器的值。

4.25 寄存器 67to72-陀螺仪测量值：GYRO\_XOUT\_H, GYRO\_XOUT\_L, GYRO\_YOUT\_H, GYRO\_YOUT\_L, GYRO\_ZOUT\_H, GYRO\_ZOUT\_L

Type: Read Only

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
43	67	GYRO_XOUT[15:8]							
44	68	GYRO_XOUT[7:0]							
45	69	GYRO_YOUT[15:8]							
46	70	GYRO_YOUT[7:0]							
47	71	GYRO_ZOUT[15:8]							
48	72	GYRO_ZOUT[7:0]							

说明：

这些寄存器存储了最近的陀螺仪测量值。陀螺仪测量值以采样率被写入这些寄存器。

温度测量寄存器，以及加速度测量值寄存器、陀螺仪测量值寄存器、外部传感器数据寄存器，都是由两套寄存器组成：一套内部寄存器和一套面向用户的只读寄存器。

传感器内部寄存器内的数据组总是以采样率的速度更新。同时，如果串口可用，那么面向用户的寄存器组复制内部寄存器组的值。这保证了对寄存器组突发读取（burst read，应该是连续地址的字节读取）时可以读取到同一采样率时刻的测量值。注意，如果突发读取没有用到，用户应该通过利用数据就绪中断保证在单字节读取时可以读取到当前采样时刻的值。

每一个陀螺仪测量值的满量程在 *FS\_SEL* (Register 27)中设置。对应的灵敏度如下表所示：

FS_SEL	Full Scale Range	LSB Sensitivity
0	± 250 °/s	131 LSB/°/s
1	± 500 °/s	65.5 LSB/°/s
2	± 1000 °/s	32.8 LSB/°/s
3	± 2000 °/s	16.4 LSB/°/s

参数：

*GYRO\_XOUT*                      16 位补码值，存储最近的 X 轴陀螺仪测量值。

*GYRO\_YOUT*                      16 位补码值，存储最近的 Y 轴陀螺仪测量值。

*GYRO\_ZOUT*                      16 位补码值，存储最近的 Z 轴陀螺仪测量值。

4.26 寄存器 73to96-外部传感器数据寄存器：EXT\_SENS\_DATA\_00 到 EXT\_SENS\_DATA\_23

#### Type: Read Only

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
49	73	EXT_SENS_DATA_00[7:0]							
4A	74	EXT_SENS_DATA_01[7:0]							
4B	75	EXT_SENS_DATA_02[7:0]							
4C	76	EXT_SENS_DATA_03[7:0]							
4D	77	EXT_SENS_DATA_04[7:0]							
4E	78	EXT_SENS_DATA_05[7:0]							
4F	79	EXT_SENS_DATA_06[7:0]							
50	80	EXT_SENS_DATA_07[7:0]							
51	81	EXT_SENS_DATA_08[7:0]							
52	82	EXT_SENS_DATA_09[7:0]							
53	83	EXT_SENS_DATA_10[7:0]							
54	84	EXT_SENS_DATA_11[7:0]							
55	85	EXT_SENS_DATA_12[7:0]							
56	86	EXT_SENS_DATA_13[7:0]							
57	87	EXT_SENS_DATA_14[7:0]							
58	88	EXT_SENS_DATA_15[7:0]							
59	89	EXT_SENS_DATA_16[7:0]							
5A	90	EXT_SENS_DATA_17[7:0]							
5B	91	EXT_SENS_DATA_18[7:0]							
5C	92	EXT_SENS_DATA_19[7:0]							
5D	93	EXT_SENS_DATA_20[7:0]							
5E	94	EXT_SENS_DATA_21[7:0]							
5F	95	EXT_SENS_DATA_22[7:0]							
60	96	EXT_SENS_DATA_23[7:0]							

描述：

这些寄存器存储辅助 I2C 总线上从设备 0、1、2、3 上挂载的外部传感器的测量数据，从设备 4 上的数据存储在 *I2C\_SLV4\_DI* (Register 53)。

外部传感器数据已采样率的速度写入这些寄存器里。访问速度可以通过修改 *Slave Delay Enable* 寄存器 (Register 103)来在采样率的基础上降低。

外部传感器数据寄存器，和陀螺仪测量寄存器、加速度寄存器、温度测量寄存器，都是有两套寄存器组成的：一套内部寄存器组和一套面向对象的寄存器组。

外部传感器的寄存器组的数据在串口可用的时候总是以采样率（或者降低的采样率）的速度更新。这保证了对寄存器组突发读取（burst read，应该是连续地址的字节读取）时可以读取到同一采样率时刻的测量值。注意，如果突发读取没有用到，用户应该通过利用数据就绪中断保证在单字节读取时可以读取到当前采样时刻的值。

数据根据寄存器 *I2C\_SLV0\_CTRL*, *I2C\_SLV1\_CTRL*, *I2C\_SLV2\_CTRL*, *I2C\_SLV3\_CTRL* (Registers 39, 42, 45, 48)的配置内容放入这些寄存器中。当从一个使能(*I2C\_SLVx\_EN = 1*)的从设备中读取多于 0 个字节(*I2C\_SLVx\_LEN > 0*)的数据的时候，从设备的数据以采样率(as defined in Register 25)或者减少后的采样率（在寄存器 52 和 103 中设置）。在每一个采样周期，从设备的读取操作是按设备号以此执行的。如果所有从设备均使能，且要读取的字节数均大于 0，那么读取的顺序为：从设备 0，从设备 1，从设备 2，从设备 3。

每一个使能的从设备都根据读取的字节长度(*I2C\_SLVx\_LEN*)以及设备号在中与有与之对应的一组寄

寄存器，且从 EXT\_SENS\_DATA\_00 开始。注意，这意味着使能或者禁止一个从设备可能会改变高编号从设备相对应的寄存器组。进一步说，如果这样改变之后本来要读取的数据中只有一部分被读取，那么那些留在已经不与某从设备有关联的寄存器中的数据在复位之前将一直留着那里。

如果从设备要读取的数据之和大于 EXT\_SENS\_DATA 寄存器的个数，超出的部分将会被丢掉。总共有 24 个 EXT\_SENS\_DATA 寄存器，因此读取从设备的数据的和不能超过 24 个字节，否则将会产生丢失。

注意：从设备 4 的行为与从设备 0-3 有区别。关于从设备 4 的更多特性，请参考寄存器 49-53。

示例：

假设从设备使能且有 4 个数据要被读取( $I2C\_SLV0\_EN = 1$  and  $I2C\_SLV0\_LEN = 4$ )，从设备 1 使能且有两个字节数据要被读取( $I2C\_SLV1\_EN=1$  and  $I2C\_SLV1\_LEN = 2$ )。在这样的条件下，寄存器 EXT\_SENS\_DATA\_00 到\_03 将会与从设备 0 关联，而 04-05 将会与从设备 1 关联。如果从设备 2 也使能，从 06 开始的寄存器将会与从设备 2 关联。如果从设备 2 禁止而从设备 3 使能，那么从 06 开始的寄存器将会与从设备 3 关联。

动态禁止时寄存器分配：

如果一个从设备一直禁止，最初分配给它的寄存器组将会一直与它相关联。这样就避免了寄存器组的动态调整。

寄存器组的分配在下面两种情况下将会从新分配：（1）所有从设备禁止（2） $I2C\_MST\_RST$ (Register 106)位被置位。

如果一个从设备收到 NACKED 信号或者停止运行，上面的也同样成立。

4.27 寄存器 97-运动检测状态：MOT\_DETECT\_STATUS

Type: Read Only

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
61	97	MOT_XNEG	MOT_XPOS	MOT_YNEG	MOT_YPOS	MOT_ZNEG	MOT_ZPOS	-	MOT_ZRMOT

描述：

这个寄存器展示了运动检测和静态运动检测的状态。

运动检测位 *MOT\_XNEG*, *MOT\_XPOS*, *MOT\_YNEG*, *MOT\_YPOS*, *MOT\_ZNEG*, *MOT\_ZPOS* 展示了触发运动检测中断的运动各个不同轴以及同轴不同极性的状态。

*MOT\_ZRMOT* 位当静态运动被检测到时置位。

读该寄存器的动作将会清除该寄存器的运动检测位。然而，在静态运动不再能被检测到之前，*MOT\_ZRMOT* 位不能被清除。

关于运动检测和静态运动检测的更多新城信息，请参考寄存器 31-34 以及《PU-6000/MPU-6050 Product Specification document》的 Section8.3 和 Section8.4。

位 1 保留。

参数：

<i>MOT_XNEG</i>	当在 X 负轴的运动触发了运动检测中断，该位被自动置位。
<i>MOT_XPOS</i>	当在 X 正轴的运动触发了运动检测中断，该位被自动置位。
<i>MOT_YNEG</i>	当在 Y 负轴的运动触发了运动检测中断，该位被自动置位。
<i>MOT_YPOS</i>	当在 Y 正轴的运动触发了运动检测中断，该位被自动置位。
<i>MOT_ZNEG</i>	当在 Z 负轴的运动触发了运动检测中断，该位被自动置位。
<i>MOT_ZPOS</i>	当在 Z 正轴的运动触发了运动检测中断，该位被自动置位。
<i>MOT_ZRMOT</i>	当静态运动检测中断被触发时，该位自动置位。

#### 4.28 寄存器 99-I2C 从设备 0 数据输出：I2C\_SLV0\_DO

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
63	99	I2C_SLV0_DO[7:0]							

描述：

当从设备 0 被设置为写模式的时候，这个寄存器保存了要写入从设备 0 的数据。关于从设备 0 控制的更多信息，请参考寄存器 37-39。

参数：

<i>I2C_SLV0_DO</i>	当从设备 0 为写模式时要写入从设备 0 的 8 位无符号数。
--------------------	---------------------------------

### 4.29 寄存器 100-I2C 从设备 1 数据输出：I2C\_SLV1\_DO

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
64	100	I2C_SLV1_DO[7:0]							

描述：

当从设备 1 被设置为写模式的时候，这个寄存器保存了要写入从设备 1 的数据。关于从设备 1 控制的更多信息，请参考寄存器 40-42。

参数：

*I2C\_SLV1\_DO*      当从设备 1 为写模式时要写入从设备 1 的 8 位无符号数。

### 4.30 寄存器 101-I2C 从设备数据输出：I2C\_SLV2\_DO

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
65	101	I2C_SLV2_DO[7:0]							

描述：

当从设备 2 被设置为写模式的时候，这个寄存器保存了要写入从设备 2 的数据。关于从设备 2 控制的更多信息，请参考寄存器 43-45。

参数：

*I2C\_SLV2\_DO*      当从设备 2 为写模式时要写入从设备 2 的 8 位无符号数。

### 4.31 寄存器 102-I2C 从设备 3 数据输出：I2C\_SLV3\_DO

**Type: Read/Write**

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
66	102	I2C_SLV3_DO[7:0]							

**描述:**

当从设备 3 被设置为写模式的时候，这个寄存器保存了要写入从设备 3 的数据。关于从设备 3 控制的更多信息，请参考寄存器 46-48。

**参数:**

*I2C\_SLV3\_DO* 当从设备 3 为写模式时要写入从设备 3 的 8 位无符号数。

**4.32 寄存器 103-I2C 主机延迟控制: I2C\_MST\_DELAY\_CTRL**

**Type: Read/Write**

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
67	103	DELAY_ES_SHADOW	-	-	I2C_SLV4_DLY_EN	I2C_SLV3_DLY_EN	I2C_SLV2_DLY_EN	I2C_SLV1_DLY_EN	I2C_SLV0_DLY_EN

**描述:**

这个寄存器用来指定外部传感器数据跟踪(data shadowing)的定时。这个寄存器也用来设置降低从设备的访问速度（在采样率的基础上）。

当位 *DELAY\_ES\_SHADOW* 被置位时，外部传感器数据追踪一直延迟知道所有数据被接收成功。

当位 *I2C\_SLV4\_DLY\_EN*, *I2C\_SLV3\_DLY\_EN*, *I2C\_SLV2\_DLY\_EN*, *I2C\_SLV1\_DLY\_EN*, *I2C\_SLV0\_DLY\_EN* 被置位时，对应的从设备的访问速度将会降低。当从设备访问速度降低时，其访问速度为:

$$1 / (1 + I2C\_MST\_DLY) \text{ samples.}$$

基础采样率的设置在 *SMPLRT\_DIV* (register 25)和 *DLPF\_CFG*(register 26)。

关于的更多信息，请参考寄存器 52。

关于采样率的更多信息，请参考寄存器 25。

位 5、6 保留。

**参数:**

*DELAY\_ES\_SHADOW* 置位时，延迟外部传感器数据追踪一直到所有数据接收成功。

*I2C\_SLV4\_DLY\_EN* 使能时，从设备 4 以降低后的采样率访问。



*I2C\_SLV3\_DLY\_EN* 使能时，从设备 3 以降低后的采样率访问。

*I2C\_SLV2\_DLY\_EN* 使能时，从设备 2 以降低后的采样率访问。

*I2C\_SLV1\_DLY\_EN* 使能时，从设备 1 以降低后的采样率访问。

*I2C\_SLV0\_DLY\_EN* 使能时，从设备 0 以降低后的采样率访问。

#### 4.33 寄存器 104-信号路径复位：SIGNAL\_PATH\_RESET

Type: Write Only

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
68	104	-	-	-	-	-	GYRO_RESET	ACCEL_RESET	TEMP_RESET

描述：

这个传感器用来重置陀螺仪、加速度计、温度传感器的模拟和数字信号路径。

该重置将会使模拟和数字转换器和滤波器回到上电时的配置。

注意:这个寄存器不会清除传感器寄存器。

位 3-7 保留。

参数：

*GYRO\_RESET* 置位时，重置陀螺仪模拟和数字信号路径。

*ACCEL\_RESET* 置位时，重置加速度计模拟和数字信号路径。

*TEMP\_RESET* 置位时，重置温度传感器模拟和数字信号路径。

#### 4.34 寄存器 105-运动检测控制：MOT\_DETECT\_CTRL

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
69	105	-	-	ACCEL_ON_DELAY[1:0]		FF_COUNT[1:0]		MOT_COUNT[1:0]	

描述：

这个寄存器可以延迟加速度计的上电时间。也可以用来配置自由落体和运动检测的渐减速度。



加速度计数据通路像传感器寄存器、运动检测、静态运动检测、自由落体检测提供数据样本。包含滤波器的数据通路一定要在运动检测中断开始之前通过新的样本唤醒。默认的唤醒延迟位 4ms，通过本寄存器可以延迟为最高 7ms。额外的延迟在 *ACCEL\_ON\_DELAY* 中配置，且 1 LSB = 1 ms。用户可以选择任何大于 0 的值，除非 InvenSense 另外指示。关于检测模型的更多信息，请参考《MPU-6000/MPU-6050 Product Specification document》的 Section8。

当加速度计的特定个数的测量值满足自由落体检测模型或者运动检测模型对应的阈值，相对应的运动检测将会被启动。当阈值条件满足时，对应的运动检测计时器会增加 1。用户可以通过配置 *FF\_COUNT* 和 *MOT\_COUNT* 来控制当阈值条件没满足时的运动检测计时器的渐减速度。渐减速度可以按下表进行设置：

FF_COUNT or MOT_COUNT	Counter Decrement
0	Reset
1	1
2	2
3	4

当 *FF\_COUNT* 或者 *MOT\_COUNT* 被置为 0（复位），任何不合格的采样将会重置对应的计数器为 0。关于自由落体检测和运动检测的更多信息，请参考寄存器 29-32。

位 6、7 保留。

参数：

<i>ACCEL_ON_DELAY</i>	2 位无符号数。指定延迟时间。1 LSB = 1 ms。
<i>FF_COUNT</i>	2 位无符号数，配置自由落体检测计数器的渐减速度。
<i>MOT_COUNT</i>	2 位无符号数，配置运动检测计数器的渐减速度。

4.35 寄存器 106-用户控制：USER\_CTRL

Type: Read/Write

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

描述：

这个寄存器允许用户使能或者禁止 FIFO 缓冲、I2C 主机模式和主 I2C 接口。FIFO、I2C 主机、传

传感器信号通路和传感器寄存器也可以用这个寄存器实现重置。

当 *I2C\_MST\_EN* 置位时，I2C 主机模式使能。在这种模式下，MPU 是作为挂在辅助 I2C 总线上的主机存在。当该位清零时，辅助 I2C 总线路线(AUX\_DA and AUX\_CL)被主 I2C 总线(SDA and SCL)逻辑驱动。这是使能旁路模式的前提。关于旁路模式的更多信息，请参考寄存器 55。

MPU-6000: 当 *I2C\_IF\_DIS* 置位时，主 SPI 接口将会使能以替代主 I2C 接口。

MPU-6050: 总是向 *I2C\_IF\_DIS* 写 0。

当重置位(*FIFO\_RESET*, *I2C\_MST\_RESET*, *SIG\_COND\_RESET*)被置位时，这些重置位将会触发复位然后这些位会被清零。

位 3、7 保留。

#### 参数:

<i>FIFO_EN</i>	置位时，该位使能 FIFO 操作。清零时，FIFO 被禁止，此时 FIFO 不能被读写。除非 MPU 重启（power cycled），否则 FIFO 的状态不会改变。
<i>I2C_MST_EN</i>	置位时，使能 I2C 主机模式。当该位清零时，辅助 I2C 总线路线(AUX_DA and AUX_CL)被主 I2C 总线(SDA and SCL)逻辑驱动。
<i>I2C_IF_DIS</i>	MPU-6000: 置位时，禁止主 I2C 接口并使能 SPI 接口。 MPU-6050: 总是写 0。
<i>FIFO_RESET</i>	当 <i>FIFO_EN</i> 位为 0 是，置位该位将会重置 FIFO，且触发重置之后该位自动清零。
<i>I2C_MST_RESET</i>	当 <i>I2C_MST_EN</i> 位为 0 时，该位置位将会复位 I2C 主机，且触发重置后该位自动清零。
<i>SIG_COND_RESET</i>	置位时，该位复位所有传感器（陀螺仪，加速度计，温度传感器）的信号通路。这个操作同时也会清除传感器寄存器。这些重置被触发后，该位自动清零。 如果要只重置信号通路，请用寄存器 104 的 <i>SIGNAL_PATH_RESET</i> 。

## 4.36 寄存器 107-电源管理 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]		

描述：

这个寄存器允许用户配置电源模式和时钟源。它还提供一个位用来重置整个设备，已经一个位用来禁止温度传感器。

通过置位 *SLEEP*，MPU 将进入低功耗睡眠模式。当置位 *CYCLE* 且睡眠模式被禁止，MPU 将会进入循环模式。在循环模式下，设备将在睡眠模式和唤醒以按照 *LP\_WAKE\_CTRL* (register 108)指定频率的寄存器从自主传感器中读取单个样本数据。配置唤醒频率要用电源管理 2 寄存器(Register 108)中的 *LP\_WAKE\_CTRL*。

内部 8MHz 振荡器、陀螺仪基础时钟、外部时钟源都可以被用来作为 MPU 的时钟源。当内部 8MHz 振荡器或者外部时钟源被选择为时钟源，MPU 可以在陀螺仪禁止的时候在低功耗模式下操作。

上电时，MPU 的时钟源默认为内部振荡器。然而，非常推荐设备配置为用陀螺仪或者外部时钟作为时钟源以提高稳定性。时钟源的选择可以参考下表：

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

参数：

<i>DEVICE_RESET</i>	置位后，把所有的内部寄存器复位到他们的默认值。当复位完成时，该位自动清零。传感器的默认值可以在 Section3 中找到。
<i>SLEEP</i>	置位后，MPU 进入睡眠模式。
<i>CYCLE</i>	置位且睡眠模式被禁止时，MPU 将会进去循环模式。
<i>TEMP_DIS</i>	置位后，禁止温度传感器。
<i>CLKSEL</i>	3 位无符号数，为设备指定时钟源。

### 4.37 寄存器 108-电源管理 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	

#### 描述:

这个寄存器允许用户配置在“只有加速度计工作低功耗模式”(Accelerometer Only Low Power Mode)下的唤醒频率。这个寄存器也允许用户加速度计和陀螺仪的单个轴进入待机模式。

MPU-9150 可以通过置位电源管理 1 寄存器(Register 107)中的 *PWRSEL* 位进入“只有加速度计工作低功耗模式”。在这种模式下，MPU 将会关闭处理主 I2C 接口的所有设备，只以固定的时间间隔唤醒加速度计以进行单个采样 (single measurement)。唤醒频率可以用 *LP\_WAKE\_CTRL* 配置，具体如下表所示：

LP_WAKE_CTRL	Wake-up Frequency
0	1.25 Hz
1	2.5 Hz
2	5 Hz
3	10 Hz

用户可以通过利用这个寄存器把加速度计和陀螺仪的单轴配置进入待机模式。如果该设备正在用陀螺仪的某个轴作为时钟源并且该轴被配置进入待机模式，设备的时钟源会自动变成内置的 8MHz 振荡器。

#### 参数:

*LP\_WAKE\_CTRL*     2 位无符号数。

*STBY\_XA*            置位时，把加速度计的 X 轴设置为待机模式。

*STBY\_YA*            置位时，把加速度计的 Y 轴设置为待机模式。

*STBY\_ZA*            置位时，把加速度计的 Z 轴设置为待机模式。

*STBY\_XG*            置位时，把陀螺仪的 X 轴设置为待机模式。

*STBY\_YG*            置位时，把陀螺仪的 Y 轴设置为待机模式。

*STBY\_ZG*            置位时，把陀螺仪的 Z 轴设置为待机模式。

### 4.38 寄存器 114and115-FIFO 计数寄存器: FIFO\_COUNT\_H and FIFO\_COUNT\_L

Type: Read Only

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
72	114	FIFO_COUNT[15:8]							
73	115	FIFO_COUNT[7:0]							

描述：

这两个寄存器保持对当前 FIFO 中样本个数的追踪。

参数：

*FIFO\_COUNT*          十六位无符号数，指明了 FIFO 中的字节数。这个数也就是可以从 FIFO 中读出的数，而它与可用的样本数一致。

4.39 寄存器 116-FIFO 读写：FIFO\_R\_W

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
74	116	FIFO_DATA[7:0]							

描述：

这个寄存器用来对 FIFO 进行读写操作。

数据按寄存器号依次写入 FIFO 中（从低到高）。如果所有 FIFO 使能标识均使能且所有的外部传感器数据寄存器（寄存器 73-96）都与一个从设备相关联，那么，寄存器 59-96 中的内容将会以采样率的速度依次写入 FIFO。

如果传感器数据寄存器（寄存器 59-96）相对应的 FIFO 使能标识都被置位(Register 35)，那么他们中的内容将会被写入 FIFO。与 I2C 从设备 3 相关联的数据寄存器的额外标识可以在寄存器 36 中找到 (*I2C\_MST\_CTRL*)。

如果 FIFO 溢出，状态位 *FIFO\_OFLOW\_INT* 自动置位。这个位在 INT\_STATUS (Register 58)中。如果 FIFO 发生溢出，旧数据将会丢失，新数据将会被写入到 FIFO 中。

如果 FIFO 是空的，那么对它进行读写操作将会返回上次读取的数据的最后一个字节直到新数据可用。用户应该检查 *FIFO\_COUNT* 以确保 FIFO 在空的时候没有被读取。

**参数：**  
*FIFO\_DATA*            对 FIFO 读取的 8 位数。

4.40 寄存器 117-我是谁：WHO\_AM\_I

Type: Read Only

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
75	117	-	WHO_AM_I[6:1]						-

**描述：**  
这个寄存器用来查证该设备的身份。该寄存器的内容是该 MPU 的 7 位 I2C 地址的高六位。MPU 地址的最低有效位是由 AD0 引脚决定的。AD0 引脚的值没有在该寄存器中出现。  
该寄存器的默认值为 0x68。  
位 0、7 保留（0 位由硬件决定）。

**参数：**  
*WHO\_AM\_I* 包含了 MPU 的 6 位 I2C 地址。上电初始化值为 110100(bit6-1)。