

CONTENTS 目录

Document Number: RM-MPU-6000A-00

Revision: 4.0

Release Date: 03/09/2012

Translation: Tan

说明：能力有限，个别地方采用原文没进行翻译，翻译有过错误与不通顺，建议查看英语原文

对于翻译错误或者有好的修改意见欢迎发送电子邮件到 592944053@qq.com，持续更新中。

在这里我要感谢我亲爱的老猪猪！

- 1 REVISION HISTORY 修订历史**
- 2 PURPOSE AND SCOPE**
- 3 REGISTER MAP 寄存器列表**
- 4 REGISTER DESCRIPTIONS 寄存器描述**
 - 4.1 REGISTERS 13 TO 16 - SELF TEST REGISTERS 自检寄存器**
 - 4.2 REGISTER 25 - SAMPLE RATE DIVIDER 采样频率分频器**
 - 4.3 REGISTER 26 - CONFIGURATION 配置**
 - 4.4 REGISTER 27 - GYROSCOPE CONFIGURATION 陀螺仪配置**
 - 4.5 REGISTER 28 - ACCELEROMETER CONFIGURATION 加速度计配置**
 - 4.6 REGISTER 31 - MOTION DETECTION THRESHOLD 运动检测阈值**
 - 4.7 REGISTER 35 - FIFO ENABLE FIFO 使能**
 - 4.8 REGISTER 36 - I2C MASTER CONTROL I2C 主机控制**
 - 4.9 REGISTERS 37 TO 39 - I2C SLAVE 0 CONTROL I2C 从机 0 控制**
 - 4.10 REGISTERS 40 TO 42 - I2C SLAVE 1 CONTROL I2C 从机 1 控制**
 - 4.11 REGISTERS 43 TO 45 - I2C SLAVE 2 CONTROL I2C 从机 2 控制**
 - 4.12 REGISTERS 46 TO 48 - I2C SLAVE 3 CONTROL I2C 从机 3 控制**
 - 4.13 REGISTERS 49 TO 53 - I2C SLAVE 4 CONTROL I2C 从机 4 控制**
 - 4.14 REGISTER 54 - I2C MASTER STATUS I2C 主机状态**
 - 4.15 REGISTER 55 - INT PIN / BYPASS ENABLE CONFIGURATION
INT 引脚/旁路有效 使能配置**
 - 4.16 REGISTER 56 - INTERRUPT ENABLE 中断使能**
 - 4.17 REGISTER 58 - INTERRUPT STATUS 中断状态**
 - 4.18 REGISTERS 59 TO 64 - ACCELEROMETER MEASUREMENTS
加速度计测量值**
 - 4.19 REGISTERS 65 AND 66 - TEMPERATURE MEASUREMENT
温度测量值**
 - 4.20 REGISTERS 67 TO 72 - GYROSCOPE MEASUREMENTS
陀螺仪测量值**
 - 4.21 REGISTERS 73 TO 96 - EXTERNAL SENSOR DATA 外部传感器数据**
 - 4.22 REGISTER 99 - I2C SLAVE 0 DATA OUT I2C 从机 0 数据输出**
 - 4.23 REGISTER 100 - I2C SLAVE 1 DATA OUT I2C 从机 1 数据输出**
 - 4.24 REGISTER 101 - I2C SLAVE 2 DATA OUT I2C 从机 2 数据输出**
 - 4.25 REGISTER 102 - I2C SLAVE 3 DATA OUT I2C 从机 3 数据输出**
 - 4.26 REGISTER 103 - I2C MASTER DELAY CONTROL I2C 主机延时管理**

- 4.27 REGISTER 104 - SIGNAL PATH RESET** 信号通道复位
- 4.28 REGISTER 105 - MOTION DETECTION CONTROL** 运动检测控制
- 4.29 REGISTER 106 - USER CONTROL** 用户配置
- 4.30 REGISTER 107 - POWER MANAGEMENT 1** 电源管理 1
- 4.31 REGISTER 108 - POWER MANAGEMENT 2** 电源管理 2
- 4.32 REGISTER 114 AND 115 - FIFO COUNT REGISTERS** FIFO 计数寄存器
- 4.33 REGISTER 116 - FIFO READ WRITE** FIFO 读写
- 4.34 REGISTER 117 - WHO AM I** 我是谁

1 REVISION HISTORY 修订历史
[请参阅原文](#)

2 PURPOSE AND SCOPE
[请参阅原文](#)

3 REGISTER MAP 寄存器列表
[请参阅原文](#)

4 REGISTER DESCRIPTIONS 寄存器描述

本节介绍 MPU-60X0 的每个寄存器的功能和内容。

注意：设备在上电时会进入睡眠模式。

4.1 REGISTERS 13 TO 16 - SELF TEST REGISTERS 自检寄存器

SELF_TEST_X, SELF_TEST_Y, SELF_TEST_Z, and SELF_TEST_A

Type: Read/Write

| Register (Hex) | Register (Decimal) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|----------------|--------------------|--------------|------|--------------|--------------|--------------|------|--------------|------|
| 0D | 13 | XA_TEST[4-2] | | | XG_TEST[4-0] | | | | |
| 0E | 14 | YA_TEST[4-2] | | | YG_TEST[4-0] | | | | |
| 0F | 15 | ZA_TEST[4-2] | | | ZG_TEST[4-0] | | | | |
| 10 | 16 | RESERVED | | XA_TEST[1-0] | | YA_TEST[1-0] | | ZA_TEST[1-0] | |

说明：
这些寄存器允许用户用于陀螺仪和加速度计的机械和电气自检。下面描述自检过程。

1.陀螺仪硬件自检：相关方法

陀螺仪允许用户测试机械和电气部分。

自检的代码在 InvenSense 提供的 MotionApps™软件里面。如果没有使用 MotionApps™软件，请参阅下节(Obtaining the Gyroscope Factory Trim (FT) Value)。

当自检启动，片上的电子设备就会启动相应的传感器。这次启动会使传感器的 proof masses 的距离相当于一个预定的科里奥利力（Coriolis force）。传感器的 Proof masses 位移变换的结果将在输出信号中反映。输出信号用户可以在自检反馈看到。

自检反馈（STR）定于如下：

Self Test Response=

Gyroscope Output with Self Test Enabled — Gyroscope Output with Self Test Disabled

自检反馈通过反馈与出产的评估比较查找变换用于确定这部分通过还是自检失败

4.2 REGISTER 25 – SAMPLE RATE DIVIDER

采样频率分频器

SMPRT_DIV

Type: Read/Write

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

说明:

该寄存器用于 MPU-60X0 的陀螺仪采样频率输出设置。

传感器寄存器输出, **FIFO** 输出, **DMP** 采样, **Motion** 检测, **Zero Motion** 检测和 **Free Fall** 检测都基于这个采样频率。

采样频率=陀螺仪输出频率/ (1+SMPLRT_DIV)

当 **DLPF is disabled** (**DLPF_CFG=0 or 7**), 陀螺输出频率=8kHz; 当 **DLPF is enabled** (see 寄存器 26), 陀螺仪输出频率=1KHz。

注意: 加速度计输出频率为 1KHz。这意味着, 对于一个大于 1kHz 的采样率, 同个加速度计的采样, 可能不止一次是输出到 **FIFO**, **DMP**, 传感器寄存器。

陀螺仪和加速度计信号路径图, 请参阅第 8 条 MPU-6000/MPU-6050 产品规格文件。

参数:

SMPLRT_DIV 8 位无符号值。陀螺仪输出频率由这个值的分频所确定。

4.3 REGISTER 26 - CONFIGURATION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:

该寄存器配置外部 **Frame Synchronization (FSYNC)** 引脚采样, 陀螺仪和加速度计的数
字低通滤波器。

通过配置 **EXT_SYNC_SET** 可以使用一个外部信号连接到 **FSYNC** 引脚进行采样。

FSYNC 引脚的信号的变化被锁存, 使短的选通信号可能被捕获。锁存 **FSYNC** 信号将
作为采样的采样频率, 定义在寄存器 **25**。采样结束后, 锁存器将复位到当前的 **FSYNC** 信
号状态。

EXT_SYNC_SET 的值确定采样的值将代替传感器数据寄存器中的最低有效位。替换如
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)由 DLPF_CFG 配置。加速度计和陀螺仪根据 DLPF_CFG 的值被
过滤。下表显示过滤情况:

| 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 |

第 6、7 位保留。

参数:

EXT_SYNC_SET 3 位无符号数值。配置 **FSYNC** 引脚采样。

DLPF_CFG 3 位无符号数值。配置 **DLPF** 设置。

4.4 REGISTER 27 - GYROSCOPE CONFIGURATION

陀螺仪配置

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] | | - | - | - |

说明：

这寄存器是用来触发陀螺仪自检和配置陀螺仪的满量程范围。

陀螺仪自检允许用户检测机械和电气部分。每个陀螺仪的轴通过控制这个寄存器的 **XG_ST**, **YG_ST**, 和 **ZG_ST** 位来激活。各轴的自检可独立进行或全部在相同的时间进行。

当自检启动，片上的电子设备就会启动相应的传感器。这次启动会使传感器的 **proof masses** 的距离相当于一个预定的科里奥利力（**Coriolis force**）。传感器的 **Proof masses** 位移变换的结果将在输出信号中反映。输出信号用户可以在自检反馈看到。

自检反馈（STR）定于如下：

Self Test Response=

Gyroscope Output with Self Test Enabled — Gyroscope Output with Self Test Disabled

每个陀螺仪轴的自检极限在 **MPU-6000/MPU-6050** 产品规格文件的电气特性表中。当反馈的值为产品的最小到最大极限之间时，表示通过自检。如果反馈的值超过文档中的最大最小特性，表示自检失败。

FS_SEL 选择陀螺仪的满量程，如下表：

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

第 0 到 2 位保留。

参数：

XG_ST 该位置位 **X** 轴进行自检。

YG_ST 该位置位 **Y** 轴进行自检。

ZG_ST 该位置位 **Z** 轴进行自检。

FS_SEL 2 位无符号数值。。选择陀螺仪的满量程范围。

4.5 REGISTER 28 - ACCELEROMETER CONFIGURATION

加速度计配置

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] | | - | | |

说明:

这寄存器是用来触发加速度计自检和配置加速度计的满量程范围。这个寄存器也可以用于配置数字高通滤波器（DHPF）。

加速度计自检允许用户检测机械和电气部分。每个加速度计的轴可以通过控制 该寄存器的 **XA_ST**，**YA_ST**和 **ZA_ST** 位来激活。每个轴的自检可以独立工作或者全部同时工作。

当自检激动时，芯片上的电路会启动传感器。这个启动时模仿一个外部的力量，启动的传感器会产生一个相应的输出信号。该输出的信号用于发现自检反馈。

自检反馈的定义如下：

自检反馈 = 传感器自检使能输出 — 自检不包含使能的输出

(Self-test response =

Sensor output with self-test enabled - Sensor output without self-test enabled)

每个轴的自检极限值在 **MPU60X0** 产品特性文档的电气特性表提供。当自检反馈的值在产品特性的最大/最小值之间，自检通过。如果自检反馈的值超出了文档中的最大最小极限，则自检失败。

AFS_SEL 用于选择加速度计的满量程范围，如下表：

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

参数：

| | |
|----------------|-----------------------------|
| XA_ST | 该位置 1，加速度计的 X 轴执行自检。 |
| YA_ST | 该位置 1，加速度计的 Y 轴执行自检。 |
| ZA_ST | 该位置 1，加速度计的 Z 轴执行自检。 |
| AFS_SEL | 2 位无符号值。 选择加速度计的满量程范围。 |

4.6 REGISTER 31 - MOTION DETECTION THRESHOLD

运动检测阈值

Type: Read/Write

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

说明:

这个寄存器用于配置运动中断产生的检测阈值。在 **MPU-6000/MPU-6050** 产品特性文档中的电子特性表可以找到 **MOT_THR** 每 **mg** 增加的最低有效位 (**LSB**)

任何的加速计测量的绝对值超过此运动检测阈值时，运动被检测到。

MOT_DETECT_STATUS (寄存器 **97**) 的运动中断可以表明检测到哪个轴和方向在运动。

运动检测中断的更多详细信息，请参阅 **MPU** 产品规格特性的第 **8.3** 节的以及本文档的寄存器 **56** 和 **58**。

参数:

MOT_THR **8** 位无符号数值。指定动作检测阈值。

4.7 REGISTER 35 - FIFO ENABLE FIFO 使能

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 |

说明:

此寄存器决定哪个传感器的测量值被加载到 **FIFO** 缓冲区。

如果这个寄存器中相关的传感器 **FIFO_EN** 位被置 **1**，传感器的数据寄存器 (寄存器 **59** 到 **96**) 存储数据就会被加载到 **FIFO** 缓冲区。

当这个寄存器中传感器的 **FIFO_EN** 位被使能，传感器的数据寄存器中数据就会被加载到 **FIFO** 缓冲区。传感器的采样频率在寄存器 **25** 中定义。更多关于传感器的数据寄存器的信息，请参阅寄存器 **59 to 96**。

当一个外部从机相应的 **FIFO_EN** 位 (**SLV_x_FIFO_EN**, **x=0, 1** 或 **2**) 被置 **1**，存储在它相应的数据寄存器 (**EXT_SENS_DATA** 寄存器, 寄存器 **73 to 96**) 中的数据会根据采样频率写入到 **FIFO** 缓冲区。**I2C** 从机的 **EXT_SENS_DATA** 寄存器由 **I2C_SLV_x_CTRL** (**x=0, 1** 或 **2**, 寄存器 **39,42** 和 **45**) 寄存器确定。更多关于 **EXT_SENS_DATA** 寄存器的信息，请参阅寄存器 **73 to 96**。

需要注意的是相应的 **FIFO_EN** 的位 (**SLV3_FIFO_EN**) 在 **I2C_MST_CTRL** (寄存器 **36**)。同时要注意的是从机 **4** 和从机 **0-30** 不同，更多关于从机 **4** 的用法请参阅寄存器 **49 to 53**。

参数:

TEMP_FIFO_EN

该位置 1, 该位使能 **TEMP_OUT_H** 和 **TEMP_OUT_L** (寄存器 65 和 66) 可以加载到 **FIFO** 缓冲区。

XG_FIFO_EN

该位置 1, 该位使能 **GYRO_XOUT_H** 和 **GYRO_XOUT_L** (寄存器 67 和 68) 可以加载到 **FIFO** 缓冲区。

YG_FIFO_EN

该位置 1, 该位使能 **GYRO_YOUT_H** 和 **GYRO_YOUT_L** (寄存器 69 和 70) 可以加载到 **FIFO** 缓冲区。

ZG_FIFO_EN

该位置 1, 该位使能 **GYRO_ZOUT_H** 和 **GYRO_ZOUT_L** (寄存器 71 和 72) 可以加载到 **FIFO** 缓冲区。

ACCEL_FIFO_EN

该位置 1, 该位使能 **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) 可以加载到 **FIFO** 缓冲区。

SLV2_FIFO_EN

该位置 1, 该位使能 **EXT_SENS_DATA** 寄存器 (寄存器 73 to 96) 和从机 2 可以加载到 **FIFO** 缓冲区。

SLV1_FIFO_EN

该位置 1, 该位使能 **EXT_SENS_DATA** 寄存器 (寄存器 73 to 96) 和从机 1 可以加载到 **FIFO** 缓冲区。

SLV0_FIFO_EN

该位置 1, 该位使能 **EXT_SENS_DATA registers** (寄存器 73 to 96) 和从机 0 可以加载到 **FIFO** 缓冲区。

注意: 更多关于 **EXT_SENS_DATA** 寄存器和从机设备的信息, 请参阅寄存器 73 to 96。

4.8 REGISTER 36 - I2C MASTER CONTROL I2C 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] | | | |

说明:

这个寄存器用于配置单主机或多主机控制时的辅助 I2C 总线。另外，这个寄存器也用于数据就绪中断（Data Ready interrupt）的延时，也用于从机 3 的数据写入到 FIFO 缓冲区的使能。该寄存器也配置辅助 I2C 主机从一个从机到下一个从机的读取，和配置 MPU-60X0 的 8MHz 的内部时钟。

多主机功能允许在同一个总线有多个 I2C 主机操作。有多主机需求的电路中，置 MULT_MST_EN 位为 1。这将增加约 30 μ A 的电流。

在多主机需求的电路中，每个主机必须经常监视 I2C 总线的状态。在获得总线仲裁之前，必须先确定其他 I2C 主机已经释放总线仲裁。当 MULT_MST_EN 置位为 1 时，MPU-60X0 的检测总线仲裁逻辑开启，总线是否可以能够检测。

当 WAIT_FOR_ES 位置 1 时，数据就绪中断（Data Ready interrupt）将会延时，直到外部传感器数据从从机设备加载到 EXT_SENS_DATA 寄存器。当数据就绪中断（Data Ready interrupt）被触发，这用于确保所有的内部传感器数据和外部传感器数据都被加载到各自的数据寄存器中。

当从机 3FIFO 使能位（SLV_3_FIFO_EN）置 1，从机 3 传感器测量的数据每次都会被加载到 FIFO 缓冲区。I2C 从机的 EXT_SENS_DATA 寄存器由 I2C_SLV3_CTRL（寄存器 48）确定。

更多关于 EXT_SENS_DATA 寄存器的信息，请参阅 寄存器 73 to 96。

从机 0，从机 1，从机 2 的 FIFO_EN 位在寄存器 35 中。

I2C_MST_P_NSR 位用于配置 I2C 主机从一个从机到下一个从机的读取。如果该位等于 0，读取之间会有一个重新启动。如果该位等于 1，开始读取后跟随一个停止信号。当一个写操作后接着一个读操作，开始写操作成功后跟随一个停止信号，然后就可以一直读取。

I2C_MST_CLK 由 4 位无符号值组成，用于配置 MPU-60X0 内部 8MHz 时钟的分频器。设置 I2C 主机时钟频率如下表所示：

| I2C_MST_CLK | I ² 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** 该位置 1, 使能多主机功能。
- WAIT_FOR_ES** 该位置 1, 数据就绪中断延时直到外部传感器数据从从机设备加载到 **EXT_SENS_DATA** 寄存器。
- SLV3_FIFO_EN** 该位置 1, 使能 **Slave 3** 的 **EXT_SENS_DATA** 寄存器可以写入到 **FIFO**。从机 **0-2** 可以在寄存器 **35** 找到相关位。
- I2C_MST_P_NSR** 控制 **I2C** 主机从一个从机到下一个从机的读取。该位为 **0** 时。读取之间会有一个重新启动。如果该位等于 **1**, 在下次读取时会有开始和结束信号。如果一个写操作后接着一个读操作, 开始和结束信号通常执行。
- I2C_MST_CLK** **4** 位无符号值。配置 **I2C** 主机时钟频率分频器。

4.9 REGISTERS 37 TO 39 - I2C SLAVE 0 CONTROL

I2C 从机 1 控制

I2C_SLV0_ADDR, I2C_SLV0_REG, and 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] | | | |

说明:

这些寄存器用于配置从机 0 的数据传输序列。从机 1,2,3 也是采用和从机 1 一样的方式。但是, 从机 4 和从机 0-3 的特点有很大的不同。更多关于从机 4 的信息, 请参阅寄存器 49 to 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。该位置 1, 传输方式为读操作。该位置 0, 传输方式为写操作。

I2C_SLV0_ADDR 用于指定从机的 I2C 地址。

从从机 0 的内部寄存器开始传输数据。该寄存器的地址由 I2C_SLV0_REG 指定。

传输的字节长度由 I2C_SLV0_LEN 指定。当传输大于 1 字节时 (I2C_SLV0_LEN > 1), 数据从 I2C_SLV0_REG 指定的地址开始连续的读 (写)。

在读模式中, 读取到结果放在最低的 EXT_SENS_DATA 寄存器中。更多关于读取结果的分配的信息, 请参阅 EXT_SENS_DATA 寄存器说明 (寄存器 73-96)。

在写模式中, I2C_SLV0_DO (寄存器 99) 的内容会写入到从机设备。

I2C_SLV0_EN 使能 I2C 从机的数据传输。只有发送字节数大于 0 (I2C_SLV0_LEN > 0) 和使能从机设备 (I2C_SLV0_EN = 1) 时, 数据的传输才被允许。

I2C_SLV0_BYTE_SW 用于配置字符对的字节交换 (byte swapping of word pairs)。当字节交换使能, 字符对的高字节和低字节将被交换。请参阅 I2C_SLV0_GRP 字符对的配对约定。当该位清 0, 从机 0 的字节的发送会按他们发送的顺序写入到 EXT_SENS_DATA 寄存器。

当 I2C_SLV0_REG_DIS 置 1, 只能进行读取或者写入数据。当该位清 0, 可以再读取或写入数据之前写入一个寄存器地址。当指定从机设备内部的寄存器地址进行发送或接收数据时, 该位必须等于 0。

I2C_SLV0_GRP 指定从寄存器收到的字符对的分组顺序。当该位清 0, 寄存器地址 0 和 1, 2 和 3 的字节是分别成对 (甚至, 奇数寄存器地址), 作为一个字符对。当该位置 1, 寄存器地址 1 和 2, 3 和 4 的字节是分别成对的, 作为一个字符对。

I2C 的数据发送由采样频率确定。相关定义在寄存器 **25**。用户必须确保每个启动的从机在每个采样频率信号内可以完成通讯。

I2C 从机访问速率可以比采样频率相对降低。降低的访问速率取决于 **I2C_MST_DLY**（寄存器 **52**）。不论从机的访问速率是否降低都由 **I2C_MST_DELAY_CTRL**（寄存器 **103**）确定。

从机的发送顺序是固定的。顺序为：从机 **0**，从机 **1**，从机 **2**，从机 **3**，从机 **4**。如果一个从机被关闭（**Disable**），那这个从机就被跳过。

每个从机的访问速率一般就是采样频率或者是降低后的频率。可以是一些从机的访问速率为采样频率，一些从机是降低后的访问速率。从机的访问顺序（从机 **0** to 从机 **4**）也遵循上述顺序。但是，降低访问速率的从机当还没到达访问周期时会被跳过。更多关于降低后的访问速率的信息，请参阅寄存器 **52**。无论从机的访问速率是采用频率还是降低的都由寄存器 **103** 中的 **Delay Enable** 位所决定。

参数：

| | |
|-------------------------|---|
| I2C_SLV0_RW | 该位置 1 ,为发送模式。该位清 0 , 为读取模式。 |
| I2C_SLV0_ADDR | 从机 0 的 7 位 I2C 地址 |
| I2C_SLV0_REG | 要发送到（接收）从机 0 内部寄存器的 8 位地址 |
| I2C_SLV0_EN | 该位置 1 ,使能从机 0 发送数据。该位清 0 ,失能发送。 |
| I2C_SLV0_BYTE_SW | 该位置 1 ,使能字节交换。字符对的高字节和低字节交换。请参阅 I2C_SLV0_GRP 的字符对的配对约定。 该位清 0 , 从机 0 收到的字符将按顺序写入到 EXT_SENS_DATA 寄存器中。 |
| I2C_SLV0_REG_DIS | 该位置 1 ,只能读取或写入数据。该位清 0 , 在读取或写入数据前要写入一个寄存器地址。 |
| I2C_SLV0_GRP | 1 位数值，指定从寄存器收到的字符对的分组顺序。该位清 0 , 寄存器地址 0 和 1 , 2 和 3 的字节是分别成对（甚至，奇数寄存器地址），作为一个字符对。当该位置 1 , 寄存器地址 1 和 2 , 3 和 4 的字节是分别成对的，作为一个字符对。 |
| I2C_SLV0_LEN | 4 位无符号数值。指定从机 0 发送字符的长度。该位清 0 , I2C_SLV0_EN 位自动置 0 。 |

字节交换例子 (Byte Swapping Example)

这个例子演示当 **I2C_SLV0_BYTE_SW = 1**, **I2C_SLV0_GRP = 0**, **I2C_SLV0_REG = 0x01**, and **I2C_SLV0_LEN = 0x4**:

1. 第一个字节从从机 0 寄存器 **0x01** 读取, 存储在 **EXT_SENS_DATA_00**。因为 **I2C_SLV0_GRP = 0**, 双字节读取, 奇寄存器地址将作为字符对配对在一起。只有一个字节读取时, 奇数寄存器地址代替偶数地址。
2. **I2C_SLV0_BYTE_SW = 1** 和 **I2C_SLV0_REG[0] = 1** 时, 第二个和第三个字节将被交换。从 **0x02** 读取的数据被存储在 **EXT_SENS_DATA_02**, 从 **0x03** 读取的数据将存储在 **EXT_SENS_DATA_01**。
3. 从地址 **0x04** 读取的最后一个字节将被存储在 **EXT_SENS_DATA_03**。因为只有一个字节需要读取, 交换不会进行。

从机访问例子 (Slave Access Example)

当从机 0 访问速率为采样频率, 从机 1 的访问速度为采样频率的一半。其他从机失能。在第一个周期从机 0 和从机 1 都将被访问。但是, 在第二个周期只有从机 0 被访问。在第三周期, 从机 0 和从机 1 都被访问。第四个周期, 只有从机 0 被访问。这样子循环下去。

4.10 REGISTERS 40 TO 42 – I2C SLAVE 1 CONTROL

I2C 从机 1 控制

I2C_SLV1_ADDR, I2C_SLV1_REG, and 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] | | | |

说明:

这个寄存器说明从机 1 的数据传送顺序。他们的功能和从机 0 寄存器对应的说明一样（寄存器 37 to 39）。

4.11 REGISTERS 43 TO 45 – I2C SLAVE 2 CONTROL

I2C 从机 2 控制

I2C_SLV2_ADDR, I2C_SLV2_REG, and 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] | | | |

说明:

这个寄存器说明从机 2 的数据传送顺序。他们的功能和从机 0 寄存器对应的说明一样（寄存器 37 to 39）。

4.12 REGISTERS 46 TO 48 – I2C SLAVE 3 CONTROL

I2C 从机 3 控制

I2C_SLV3_ADDR, I2C_SLV3_REG, and 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] | | | |

说明:

这个寄存器说明从机 3 的数据传送顺序。他们的功能和从机 0 寄存器对应的说明一样（寄存器 37 to 39）。

4.13 REGISTERS 49 TO 53 – I2C SLAVE 4 CONTROL

I2C 从机 4 控制

I2C_SLV4_ADDR, I2C_SLV4_REG, I2C_SLV4_DO, I2C_SLV4_CTRL, and 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] | | | | | | | |

说明：

这些寄存器说明从机 4 的数据传输顺序。从机 4 和从机 0–3 的特点有很大不同。更多关于从机 0–3 的特点信息请参阅寄存器 37 to 48。

MPU-60X0 和从机 4 的 I2C 数据传输通过设置 I2C_SLV4_RW 位控制读取或写入操作。该位置 1，传输为读取操作，该位清 0，为写入操作。

I2C_SLV4_ADDR 用于指定从机 4 的 I2C 从机地址。

I2C_SLV4_REG 用于指定数据传输从从机 4 的哪个内部寄存器开始。

在读取模式，读取结果将在 I2C_SLV4_DI 中。在写入模式，I2C_SLV4_DO 的内容将被写入从机设备。

只有 I2C_SLV4_EN 位为 1 时，数据传输才被允许。数据传输必须设置_ADDR and _REG 寄存器的参数。写入时，_DO 寄存器也需要写入内容。在传输执行结束后，I2C_SLV4_EN 位将被清 0。

如果中断使能，从机 4 数据传输完成将产生中断。寄存器 54 有该中断的状态。

当 I2C_SLV4_REG_DIS 置 1，读取或写入的数据将替代寄存器地址。当指定数据传输的从机设备内部寄存器地址后，该位等于 0。

I2C_MST_DLY 配置 I2C 从机降低后的访问速率。当从机的访问速率相对于采样频率减少，从机将在每个 $1 / (1 + I2C_MST_DLY)$ samples 访问。

这个基本采样频率由 SMPLRT_DIV（寄存器 25）和 DLPF_CFG（寄存器 26）决定。但从机的访问速率都是由 I2C_MST_DELAY_CTRL（寄存器 103）确定相对于采样频率的降低量。

更多关于采样频率的信息，请参阅寄存器 25。

参数:

| | |
|-------------------------|---|
| I2C_SLV4_RW | 该位置 1,数据传输为读取操作。 该位清 0, 数据传输为写入操作。 |
| I2C_SLV4_ADDR | 从机 4 的 7 位地址。 |
| I2C_SLV4_REG | 从机 4 数据开始传输的 8 位寄存器地址。 |
| I2C_SLV4_DO | 这个寄存器的内容会被写入从机 0。 如果 I2C_SLV4_RW 为 1(读取), 这个寄存器没有影响。 |
| I2C_SLV4_EN | 该位置 1, 使能从机 4 数据传输。 该位清 0, 失能从机 4 数据传输。 |
| I2C_SLV4_INT_EN | 该位置 1, 该位使能在从机 4 传输完成后产生一个中断信号。 该位清 0, 该位失能在从机 4 传输完成后产生一个中断信号 |
| I2C_SLV4_REG_DIS | 该位置 1, 传输将读取或写入数据。 该位清 0, 传输将读取或写入地址。 |
| I2C_MST_DLY | 配置从机设备相对于采样频率减少的访问速率。 |
| I2C_SLV4_DI | 这个寄存器存储从从机读取的数据。 数据在读取传输后填入。 |

4.14 REGISTER 54 - I2C MASTER STATUS

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:

这个寄存器展示 MPU-60X0 内部 I2C 主机产生的中断信号状态。这个寄存器也连接到主处理器的 FSYNC 中断状态。

读取这个寄存器会清除该寄存器的所有标志位。

参数:

PASS_THROUGH 该位反映从外部设备产生 FSYNC 中断状态到 MPU-60X0。这作为一种方式传递一个中断从主应用处理器到 MPU-60X0。该位置 1，如果 INT_PIN_CFG 中的 FSYNC_INT_EN 置 1，这会产生一个中断。(寄存器 55)。

I2C_SLV4_DONE 当从机 4 传输完成该位自动置 1。如果 INT_ENABLE 寄存器中的 I2C_MST_INT_EN 置 1(寄存器 56)和 I2C_SLV4_CTRL(寄存器 52)寄存器中的 SLV_4_DONE_INT 位置 1，则触发中断。

I2C_LOST_ARB 当 I2C 主机丢失辅助 I2C 总线(一个错误条件)仲裁，该位自动置 1。如果 INT_ENABLE 寄存器(寄存器 56)中的 I2C_MST_INT_EN 位置 1，触发一个中断。

I2C_SLV4_NACK 当从机 4 在传输中 I2C 接收到一个不应答信号 (NACK)，该位自动置 1。如果 INT_ENABLE 寄存器 (寄存器 56)中的 I2C_MST_INT_EN 位置 1，触发一个中断。

I2C_SLV3_NACK 当从机 3 在传输中 I2C 接收到一个不应答信号 (NACK)，该位自动置 1。如果 INT_ENABLE 寄存器(寄存器 56)中的 I2C_MST_INT_EN 位置 1，触发中断。

I2C_SLV2_NACK 当从机 2 在传输中 I2C 接收到一个不应答信号 (NACK)，该位自动置 1。如果 INT_ENABLE 寄存器(寄存器 56)中的 I2C_MST_INT_EN 位置 1，触发中断。

I2C_SLV1_NACK 当从机 1 在传输中 I2C 接收到一个不应答信号 (NACK)，该位自动置 1。如果 INT_ENABLE 寄存器(寄存器 56)中的 I2C_MST_INT_EN 位置 1，触发中断。

I2C_SLV0_NACK 当从机 0 在传输中 I2C 接收到一个不应答信号 (NACK)，该位自动置 1。如果 INT_ENABLE 寄存器(寄存器 56)中的 I2C_MST_INT_EN 位置 1，触发一个中断。

4.15 REGISTER 55 - INT PIN / BYPASS ENABLE CONFIGURATION

INT 引脚/旁路有效 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 | - |

说明:

这个寄存器配置 INT 引脚中断信号的产生。也用于使能 FSYNC 引脚作为主应用处理器的中断, 也使能 I2C 主机的 Bypass 模式。该位也使能时钟输出。

FSYNC_INT_EN 使能 FSYNC 引脚作为主应用处理器中断。FSYNC_INT_LEVEL 指定的过渡有效电平将触发中断。中断状态可以从 I2C 主机状态寄存器 (寄存器 54) PASS_THROUGH 位读取。

当 I2C_BYPASS_EN 等于 1 且 I2C_MST_EN (寄存器 106 第 5 位) 等于 0, 主应用处理器可以直接访问 MPU-60X0 的辅助 I2C 总线。当该位等于 0, 不管 I2C_MST_EN 什么状态, 主应用处理器都不能直接访问 MPU-60X0 的辅助 I2C 总线。

更多关于 Bypass 模式的信息, 请参阅 MPU-6000/6050 产品特性文档第 7.11 和 7.13 节。

参数:

INT_LEVEL

该位等于 0, INT 引脚的逻辑电平为高电平。
该位等于 1, INT 引脚的逻辑电平为低电平。

INT_OPEN

该位等于 0, the INT pin is configured as push-pull.
该位等于 1, the INT pin is configured as open drain.

LATCH_INT_EN

该位等于 0, INT 引脚产生 50us 脉冲。
该位等于 1, INT 引脚保持高电平直到中断被清除。

INT_RD_CLEAR

该位等于 0, 读取 INT_STATUS (寄存器 58)中断状态位清除。
该位等于 1, 任何读取中断状态位清除。

FSYNC_INT_LEVEL

该位等于 0, FSYNC 引脚的逻辑电平为高(作为主处理器中断)。
该位等于 1, FSYNC 引脚的逻辑电平为低(作为主处理器中断)。

FSYNC_INT_EN

该位等于 0, 该位失能 FSYNC 引脚从主处理器产生中断。
该位等于 1, 该位使能 FSYNC 引脚作为主处理器中断。

I2C_BYPASS_EN

该位等于 1 且 I2C_MST_EN (寄存器 106 bit[5]) 等于 0, 主应用处理器可以直接访问 MPU-60X0 的辅助 I2C 总线。
该位等于 0, 无论 I2C_MST_EN (Register 106 bit[5])什么状态, 主应用处理器不可以直接访问 MPU-60X0 的辅助 I2C 总线。

4.16 REGISTER 56 - INTERRUPT ENABLE

中断使能

INT_ENABLE

Type: Read/Write

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

说明：
该寄存器使能中断源的中断产生。

关于各个中断源产生中断的标志，请参阅寄存器 58。更多关于 I2C 主机中断的产生，请参阅寄存器 54。

位 2 和 1 保留。

参数：

- MOT_EN

该位置 1，该位使能运动检测（Motion detection）产生中断。
- FIFO_OFLOW_EN

该位置 1，该位使能 FIFO 缓冲区溢出产生中断。
- I2C_MST_INT_EN

该位置 1，该位使能 I2C 主机所有中断源产生中断。
- DATA_RDY_EN

该位置 1，该位使能数据就绪中断（Data Ready interrupt），所有的传感器寄存器写操作完成时都会产生。

4.17 REGISTER 58 - INTERRUPT STATUS

中断过状态

INT_STATUS

Type: Read Only

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

说明：
这个寄存器展示各个中断源的中断标志。每个位在读取后被清除。

关于相应的中断使能位的信息，请参阅寄存器 56。

关于 I2C 主机中断的列表，请参阅寄存器 54。
位 2 和 1 保留。

- 参数：
- MOT_INT

当运动检测（Motion Detection）中断产生，该位自动置 1。
读取寄存器该位清 0。
- FIFO_OFLOW_INT

当 FIFO 缓冲区溢出中断产生，该位自动置 1。
读取寄存器该位清 0。
- I2C_MST_INT

当 I2C 主机中断产生，该位自动置 1。
关于 I2C 主机中断的列表，请参阅寄存器 54。
读取寄存器该位清 0。
- DATA_RDY_INT

当数据就绪中断（Data Ready interrupt）产生，该位自动置 1。
读取寄存器该位清 0。

4.18 REGISTERS 59 TO 64 - ACCELEROMETER MEASUREMENTS

加速度计测量值

ACCEL_XOUT_H, ACCEL_XOUT_L, ACCEL_YOUT_H, ACCEL_YOUT_L, ACCEL_ZOUT_H, and 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] | | | | | | | |

说明:

这个寄存器存储最近加速度计的测量值。

加速度计根据采样频率（由寄存器 25 定义）写入到这些寄存器。

加速度计测量值寄存器和温度测量值寄存器，陀螺仪测量值寄存器，外部传感器数据寄存器都是由 2 个寄存器集合组成：一个内部寄存器集合和一个面向用户的读取寄存器集合。

加速度计传感器的内部寄存器集合里的数据根据采样频率更新。以此同时，每当串行接口处于闲置状态，面向用户的读取寄存器集合会复制内部寄存器集合的数据值。这保证了突发读取时传感器寄存器可以读到相同的采样时刻的测量值。请注意，如果没有突发读取，用户通过检测数据就绪中断（Data Ready interrupt）确保一组单字节的读取在相应的采样时刻。

每个 16 位加速度计测量值的满量程定义在 ACCEL_FS（寄存器 28）。对于每个满量程的设置，ACCEL_xOUT 里加速度计测量值的灵敏度最低分辨率（LSB）如下表：

| AFS_SEL | Full Scale Range | LSB Sensitivity |
|---------|------------------|-----------------|
| 0 | ±2g | 16384 LSB/g |
| 1 | ±4g | 8192 LSB/g |
| 2 | ±8g | 4096 LSB/g |
| 3 | ±16g | 2048 LSB/g |

参数:

ACCEL_XOUT 由 2 部分组成的 16 位数值
存储最近 X 轴加速度计的测量值。

ACCEL_YOUT 由 2 部分组成的 16 位数值
存储最近 Y 轴加速度计的测量值。

ACCEL_ZOUT 由 2 部分组成的 16 位数值
存储最近 Z 轴加速度计的测量值。

4.19 REGISTERS 65 AND 66 - TEMPERATURE MEASUREMENT

温度测量值

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] | | | | | | | |

说明:

这些寄存器存储最近温度传感器的测量值。

温度测量值根据采样频率（在寄存器 **25** 中定义）写到这些寄存器。

温度测量值寄存器和加速度计测量值寄存器，陀螺仪测量值寄存器，外部传感器数据寄存器都是由 **2** 个寄存器集合组成：一个内部寄存器集合和一个面向用户的读取寄存器集合。

温度传感器的内部寄存器集合里的数据根据采样频率更新。以此同时，每当串行接口处于闲置状态，面向用户的读取寄存器集合会复制内部寄存器集合的数据值。这保证了突发读取时传感器寄存器可以读到相同的采样时刻的测量值。请注意，如果没有突发读取，用户通过检测数据就绪中断（**Data Ready interrupt**）确保一组单字节的读取在相应的采样时刻。

温度传感器的比例因子和偏移在电气特性表中（**MPU-60X0** 产品特性文档中第 **6.4** 节）。

摄氏度的温度可以用寄存器的值这么计算：

Temperature in degrees C = (TEMP_OUT Register Value as a signed quantity)/340 + 36.53

请注意，在上面的方程的数学的是十进制。

参数:

TEMP_OUT

16 位有符号数值。

存储最近的温度传感器测量值。

4.20 REGISTERS 67 TO 72 - GYROSCOPE MEASUREMENTS

陀螺仪测量值

GYRO_XOUT_H, GYRO_XOUT_L, GYRO_YOUT_H, GYRO_YOUT_L, GYRO_ZOUT_H, and 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:

这个寄存器存储最近陀螺仪的测量值。

陀螺仪根据采样频率（由寄存器 25 定义）写入到这些寄存器。

陀螺仪测量值寄存器和温度测量值寄存器，加速度计测量值寄存器，外部传感器数据寄存器都是由 2 个寄存器集合组成：一个内部寄存器集合和一个面向用户的读取寄存器集合。

陀螺仪传感器的内部寄存器集合里的数据根据采样频率更新。以此同时，每当串行接口处于闲置状态，面向用户的读取寄存器集合会复制内部寄存器集合的数据值。这保证了突发读取时传感器寄存器可以读到相同的采样时刻的测量值。请注意，如果没有突发读取，用户通过检测数据就绪中断（**Data Ready interrupt**）确保一组单字节的读取在相应的采样时刻。

每个 16 位陀螺仪测量值的满量程定义在 **FS_SEL**（寄存器 27）。对于每个满量程的设置，**GYRO_xOUT** 里陀螺仪测量值的灵敏度最低分辨率（**LSB**）如下表：

| FS_SEL | Full Scale Range | LSB Sensitivity |
|--------|-------------------------------------|-------------------------------|
| 0 | $\pm 250\text{ }^{\circ}/\text{s}$ | 131 LSB/ $^{\circ}/\text{s}$ |
| 1 | $\pm 500\text{ }^{\circ}/\text{s}$ | 65.5 LSB/ $^{\circ}/\text{s}$ |
| 2 | $\pm 1000\text{ }^{\circ}/\text{s}$ | 32.8 LSB/ $^{\circ}/\text{s}$ |
| 3 | $\pm 2000\text{ }^{\circ}/\text{s}$ | 16.4 LSB/ $^{\circ}/\text{s}$ |

参数:

GYRO_XOUT 由 2 部分组成的 16 位数值
存储最近 X 轴陀螺仪的测量值。

GYRO_YOUT 由 2 部分组成的 16 位数值
存储最近 Y 轴陀螺仪的测量值。

GYRO_ZOUT 由 2 部分组成的 16 位数值
存储最近 Z 轴陀螺仪的测量值。

4.21 REGISTERS 73 TO 96 – EXTERNAL SENSOR DATA

外部传感器数据

EXT_SENS_DATA_00 through 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] | | | | | | | |

说明：

这些寄存器存储通过辅助 I2C 接口，来自从机 0—3 外部传感器的数据。从机读取的数据存储在 I2C_SLV4_DI（寄存器 53）。

外部传感器数据写入到这些寄存器的速率在寄存器 25 定义。访问速率可以通过 Slave Delay Enable 寄存器（寄存器 103）降低。

外部传感器数据寄存器和陀螺仪测量值寄存器，加速度计测量值寄存器，外部传感器数据寄存器都是由 2 个寄存器集合组成：一个内部寄存器集合和一个面向用户的读取寄存器集合。

外部传感器的内部寄存器集合里的数据根据采样频率更新。以此同时，每当串行接口处于闲置状态，面向用户的读取寄存器集合会复制内部寄存器集合的数据值。这保证了突发读取时传感器寄存器可以读到相同的采样时刻的测量值。请注意，如果没有突发读取，用户通过检测数据就绪中断（Data Ready interrupt）确保一组单字节的读取在相应的采样时刻。

数据根据 I2C_SLV0_CTRL, I2C_SLV1_CTRL, I2C_SLV2_CTRL, and I2C_SLV3_CTRL (寄存器 39, 42, 45, and 48) 放置在这些外部传感器数据寄存器的相应位置。当一个使能的从机（I2C_SLVx_EN = 1）读取多于 0 个字节（I2C_SLVx_LEN > 0）的数据，读取速率为采样频率（在寄存器 25 中定义）或者降低的速率（在寄存器 52 和 103 指定）。在每个采样周期，读取按从机的序号读取。如果所有的从机都使能且有多个字节需要读取，顺序为从机 0，从机 1，从机 2，从机 3。

Each enabled slave will have EXT_SENS_DATA registers associated with it by number of bytes read (I2C_SLVx_LEN) in order of slave number, starting from EXT_SENS_DATA_00. Note that this means enabling or disabling a slave may change the higher numbered slaves' associated registers.

Furthermore, if fewer total bytes are being read from the external sensors as a result of such a change, then the data remaining in the registers which no longer have an associated slave device (i.e. high numbered registers) will remain in these previously allocated registers unless reset.

If the sum of the read lengths of all SLVx transactions exceed the number of available EXT_SENS_DATA registers, the excess bytes will be dropped. There are 24 EXT_SENS_DATA registers and hence the total read lengths between all the slaves cannot be greater than 24 or some bytes will be lost.

Note: Slave 4's behavior is distinct from that of Slaves 0-3. For further information regarding the characteristics of Slave 4, please refer to Registers 49 to 53.

Example:

Suppose that Slave 0 is enabled with 4 bytes to be read (I2C_SLV0_EN = 1 and I2C_SLV0_LEN = 4) while Slave 1 is enabled with 2 bytes to be read, (I2C_SLV1_EN=1 and I2C_SLV1_LEN = 2). In such a situation, EXT_SENS_DATA_00 through _03 will be associated with Slave 0, while EXT_SENS_DATA_04 and 05 will be associated with Slave 1. If Slave 2 is enabled as well, registers starting from EXT_SENS_DATA_06 will be allocated to Slave 2.

If Slave 2 is disabled while Slave 3 is enabled in this same situation, then registers starting from EXT_SENS_DATA_06 will be allocated to Slave 3 instead.

Register Allocation for Dynamic Disable vs. Normal Disable

If a slave is disabled at any time, the space initially allocated to the slave in the register, will remain associated with that slave. This is to avoid dynamic adjustment.

The allocation of the EXT_SENS_DATA registers is recomputed only when disabled, or (2) the I2C_MST_RST bit is set (Register 106).

This above is also true if one of the slaves gets NACKed and stops functioning.

4.22 REGISTER 99 – I2C SLAVE 0 DATA OUT

I2C 从机 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] | | | | | | | |

说明:

当从机 0 设置位写入模式时，这个寄存器拥有输出的数据用于写入到从机 0。

更多关于从机 0 控制的信息，请参阅寄存器 37 to 39。

参数:

I2C_SLV0_DO 8 位无符号数值。当从机 0 设置为写入模式时，用于写入到从机 0。

4.23 REGISTER 100 – I2C SLAVE 1 DATA OUT

I2C 从机 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] | | | | | | | |

说明:

当从机 1 设置位写入模式时，这个寄存器拥有输出的数据用于写入到从机 1。

更多关于从机 1 控制的信息，请参阅寄存器 40 to 42。

参数:

I2C_SLV1_DO 8 位无符号数值。当从机 1 设置为写入模式时，用于写入到从机 1。

4.24 REGISTER 101 – I2C SLAVE 2 DATA OUT

I2C 从机 2 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] | | | | | | | |

说明:

当从机 2 设置位写入模式时，这个寄存器拥有输出的数据用于写入到从机 2。

更多关于从机 2 控制的信息，请参阅寄存器 43 to 45。

参数:

I2C_SLV2_DO 8 位无符号数值。当从机 2 设置为写入模式时，用于写入到从机 2。

4.25 REGISTER 102 – I2C SLAVE 3 DATA OUT

I2C 从机 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] | | | | | | | |

说明:

当从机 3 设置位写入模式时，这个寄存器拥有输出的数据用于写入到从机 3。

更多关于从机 3 控制的信息，请参阅寄存器 46 to 48。

参数:

I2C_SLV3_DO 8 位无符号数值。当从机 3 设置为写入模式时，用于写入到从机 3。

4.26 REGISTER 103 – I2C MASTER DELAY CONTROL

I2C 主机延时管理

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 |

说明：
这个寄存器用于指定外部传感器数据定时的屏蔽。也用于从机设备根据采样频率降低访问速率。

当 DELAY_ES_SHADOW 置 1，外部传感器数据屏蔽延迟直到所有数据被接收。

当 I2C_SLV4_DLY_EN, I2C_SLV3_DLY_EN, I2C_SLV2_DLY_EN, I2C_SLV1_DLY_EN, and I2C_SLV0_DLY_EN 都使能，相应的从机设备的访问速率将被降低。

当从机的访问速率根据采样频率被降低，从机将在每个 $1 / (1 + I2C_MST_DLY)$ samples 被访问。
基础采样频率由 SMPLRT_DIV(寄存器 25) and DLPF_CFG (寄存器 26)决定。

更多关于 I2C_MST_DLY 的信息，请参阅寄存器 52。
更多关于 Sample Rate 的信息，请参阅寄存器 25。
位 6 and 5 保留。

- 参数：
- DELAY_ES_SHADOW** 该位置 1，外部传感器数据延时屏蔽直到所有数据已接收。
- I2C_SLV4_DLY_EN** 该位置 1，从机 4 只能在低速率访问。
- I2C_SLV3_DLY_EN** 该位置 1，从机 3 只能在低速率访问。
- I2C_SLV2_DLY_EN** 该位置 1，从机 2 只能在低速率访问。
- I2C_SLV1_DLY_EN** 该位置 1，从机 1 只能在低速率访问。
- I2C_SLV0_DLY_EN** 该位置 1，从机 0 只能在低速率访问。

4.27 REGISTER 104 - SIGNAL PATH RESET

信号通道复位

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 |

说明：
这个寄存器用于陀螺仪，加速度计，温度传感器的模拟和数字信号通道的复位。

复位会还原模数转换信号通道和清除他们的上电配置。

注意：这个寄存器不会清除传感器寄存器。
位 7 to 3 保留。

- 参数：
- GYRO_RESET

该位置 1，复位陀螺仪的模拟和数字信号通道。
- ACCEL_RESET

该位置 1，复位加速度计的模拟和数字信号通道。
- TEMP_RESET

该位置 1，复位温度传感器的模拟和数字信号通道。

4.28 REGISTER 105 – MOTION DETECTION CONTROL

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] | | - | | - | |

说明：
这寄存器是用来添加加速度计上电的延迟时间。它也可以用来配置运动检测减少率。

加速度数据通道提供采样给传感器寄存器和运动检测模块。信号通道中新的采样必须在检测模块被唤醒开始工作之前刷新。默认的唤醒延迟，可以加长为 **3ms** 到 **4ms**。另外 **ACCEL_ON_DELAY** 中每个最低有效位 = **1ms**。用户可以选择任何大于 **0** 的数值除了 **InvenSense** 特别说明不行的。

更多关于检测模块的信息，请参阅 **MPU60X0** 产品特性文档中的第 **8** 节。
位 **7** 和 **6** 保留。

参数：
ACCEL_ON_DELAY **2** 位无符号数值。指定加速度计数据通道模块另外的上电延时。
每个最低有效位 = **1ms**。

4.29 REGISTER 106 - USER CONTROL 用户配置

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 |

说明：
这个寄存器允许用户使能或使能 **FIFO** 缓冲区，**I2C** 主机模式和主要 **I2C** 接口。**FIFO** 缓冲区，**I2C** 主机，传感器信号通道和传感器寄存器也可以使用这个寄存器复位。

当 **I2C_MST_EN** 置 1，**I2C** 主机模式使能。在这个模式下，**MPU-60X0** 作为 **I2C** 主机通过辅助 **I2C** 总线连接外部传感器从机设备。当该位清 0，辅助 **I2C** 总线(**AUX_DA** 和 **AUX_CL**)逻辑上由主 **I2C** 总线 (**SDA** 和 **SCL**) 驱动。这是使能盘路模式 (**Bypass Mode**) 的先决条件。更多关于旁路模式的信息，请参阅寄存器 55。

MPU-6000: 当 **I2C_IF_DIS** 置 1，主 **SPI** 接口的使能取代主 **I2C** 接口。

MPU-6050: 一般 **I2C_IF_DIS** 写 0。

当复位位(reset bits) (**FIFO_RESET**, **I2C_MST_RESET**, and **SIG_COND_RESET**) 置 1，这些复位位会产生一个复位并清 0。

参数:

- FIFO_EN**

该位置 1，使能 **FIFO** 操作。
该位清 0，失能 **FIFO** 缓冲区。 **FIFO** 缓冲区不能读取或写入。
FIFO 缓冲区状态不能改变除非 **MPU-60X0** 重新上电。
- I2C_MST_EN**

该位置 1，使能 **I2C** 主机模式。
该位清 0，辅助 **I2C** 总线 (**AUX_DA** 和 **AUX_CL**) 逻辑上由主 **I2C** 总线 (**SDA** 和 **SCL**) 驱动。
- I2C_IF_DIS**

MPU-6000:该位置 1，失能主 **I2C** 接口并使能 **SPI** 接口
MPU-6050: 一般该位写 0。
- FIFO_RESET**

当 **FIFO_EN=0** 该位置 1，复位 **FIFO** 缓冲区。复位后该位自动清 0。
- I2C_MST_RESET**

当 **I2C_MST_EN=0** 该位置 1，复位 **I2C** 主机。复位后该位自动清 0。
- SIG_COND_RESET**

该位置 1，复位所有传感器的信号通道 (陀螺仪，加速度计和温度传感器)。这个操作会清除传感器寄存器。复位后该位自动清 0。如果只想复位信号通道(不复位传感器寄存器)，
请使用寄存器 104， **SIGNAL_PATH_RESET**。

4.30 REGISTER 107 – POWER MANAGEMENT 1 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] | | |

说明：
这个寄存器允许用户配置电源模式和时钟源。还提供了复位整个设备和禁用温度传感器的位。

当置 **SLEEP** 位为 **1**，**MPU-60X0** 可以进入低功耗睡眠模式。当失能 **SLEEP** 且 **CYCLE** 位置 **1**，**MPU-60X0** 进入循环模式（**Cycle Mode**）。在循环模式，设备在睡眠模式和唤醒模式间循环，根据 **LP_WAKE_CTRL**（寄存器 **108**）设定的速率从加速度计采集样品数据。唤醒频率的配置，在 **Power Management 2** 寄存器（寄存器 **108**）中的 **LP_WAKE_CTRL** 位。

内部 **8MHz** 的振荡器，作为陀螺仪的基准时钟，或者是选择外部时钟作为 **MPU-60X0** 的时钟源。当内部 **8MHz** 的振荡器或者外部时钟源被选择作为时钟源，**MPU-60X0** 开始在低功耗模式下工作，陀螺仪处于失能状态。

上电后，**MPU-60X0** 的时钟源默认是内部振荡器。不管怎么样，强烈建议设备配置它（或者使用外部时钟源）作为陀螺仪的基准时钟，以提高稳定性。时钟源可以根据下表选择：

| 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 |

更多关于 **MPU-60X0** 时钟源的信息，请参阅 **MPU-60X0** 产品特性文档。
第 **4** 位保留。

- 参数：
- DEVICE_RESET** 该位置 **1**，重启内部寄存器到默认值。
复位完成后该位自动清 **0**。 各个寄存器的默认值在第 **3** 节。
- SLEEP** 该位置 **1**，**MPU-60X0** 进入睡眠模式。
- CYCLE** 当失能 **SLEEP** 且 **CYCLE** 位置 **1**，**MPU-60X0** 进入循环模式。在循环模式，设备在睡眠模式和唤醒模式间循环，根据 **LP_WAKE_CTRL**（寄存器 **108**）设定的速率从加速度计采集样品数据。
- TEMP_DIS** 该位置 **1**，失能温度传感器。
- CLKSEL** **3** 位无符号数值。指定设备的系统时钟源。

4.31 REGISTER 108 – POWER MANAGEMENT 2 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 |

说明：
这个寄存器允许用户配置加速度计在低功耗模式下唤起的频率。也允许用户让加速度计和陀螺仪的个别轴进入待机模式。

只让 MPU-60X0 的加速度计进入低功耗模式的步骤如下：

- 1.置 CYCLE 位为 1
- 2.置 SLEEP 位为 1
- 3.置 TEMP_DIS 位为 1
- 4.置 STBY_XG,STBY_YG,STBY_ZG 位为 1

上述所有位可以在电源管理 1 寄存器（寄存器 107）中找到。

在这种模式下，设备会关闭除了主 I2C 接口外其他所有设备，加速度计只在固定的间隔唤醒并测量一次。唤醒的频率可以通过配置 LP_WAKE_CTRL 实现如下：

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

更多关于 MPU-6050 电源模式的信息，请参阅寄存器 107.

用户可以使用这个寄存器实现加速度计和陀螺仪的个别轴进入待机模式。如果设备使用陀螺仪的一个轴作为时钟源，但同时这个轴进入待机模式，则时钟源会自动切换到内部 8MHz 振荡器。

参数：

| | |
|--------------|----------------------------------|
| LP_WAKE_CTRL | 2 位无符号数值。 指定加速度计在低功耗模式下的唤醒频率。 |
| STBY_XA | 该位置 1，加速度计的 X 轴进入待机模式。 |
| STBY_YA | 该位置 1，加速度计的 Y 轴进入待机模式。 |
| STBY_ZA | 该位置 1，加速度计的 Z 轴进入待机模式。 |
| STBY_XG | 该位置 1，陀螺仪的 X 轴进入待机模式。 |
| STBY_YG | 该位置 1，陀螺仪的 Y 轴进入待机模式。 |
| STBY_ZG | 该位置 1，陀螺仪的 Z 轴进入待机模式。 |

4.32 REGISTER 114 AND 115 - FIFO COUNT REGISTERS

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] | | | | | | | |

说明：

这些寄存器记录目前在 FIFO 缓冲区中的样本数。

这些寄存器映射 FIFO 计数的值。当 FIFO_COUNT_H（寄存器 72）被读取，两个寄存器都加载当前样本数。

注意：只读 FIFO_COUNT_L 并不更新当前采样数到寄存器。FIFO_COUNT_H 必须先访问来更新 2 个寄存器的内容。

FIFO_COUNT 必须按从高到低的顺序读取，确保读取到最新的 FIFO 计数值。

参数：

FIFO_COUNT 16 位无符号数值。表示存储在 FIFO 缓冲区的字节数。反过来说就是 FIFO 缓冲区中可以被读取到的字节数，他直接表示一组传感器获取的数据存储在 FIFO（寄存器 35 和 36）中可用的样本数量。

4.33 REGISTER 116 - FIFO READ WRITE 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] | | | | | | | |

说明：
这个寄存器用于 **FIFO** 缓冲区的读取和写入。

数据写入到 **FIFO** 的顺序按寄存器的序号（从最低到最高）。如果所有的 **FIFO** 使能标志（见下文）都启用，所有外部传感器数据寄存器（寄存器 **73 to 96**）有关的设备和寄存器 **56** 到 **96** 的内容都要按采样频率写入。

当传感器数据寄存器对应的 **FIFO** 使能标志位 **FIFO_EN**（寄存器 **35**）都置 **1**，传感器数据寄存器（寄存器 **59 to 96**）的内容将写入到 **FIFO** 缓冲区。另外 **I2C** 从机 **3** 传感器数据寄存器标志在 **I2C_MST_CTRL**（寄存器 **36**）。

如果 **FIFO** 缓冲区溢出，状态位 **FIFO_OFLOW_INT** 自动置 **1**。该位在 **INT_STATUS**（寄存器 **58**）。当 **FIFO** 缓冲区溢出，最老的数据就会丢失，新的数据会写入到 **FIFO**。

如果 **FIFO** 缓冲为空，读取该寄存器会返回以前从 **FIFO** 读到的最后字节，直到新的可用数据出现。用户可以检查 **FIFO_COUNT** 确保 **FIFO** 是否为空。

参数：
FIFO_DATA **8** 位数据，发送到 **FIFO** 缓冲区或者从 **FIFO** 缓冲区接收

4.34 REGISTER 117 - WHO AM I 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] | | | | | | - |

说明：这个寄存器用于标识设备的身份。**WHO_AM_I** 的内容是 **MPU-60X0** 的 7 位 **I2C** 地址的头 6 位。最后一位地址由 **AD0** 引脚确定。**AD0** 引脚的值跟寄存器无关。

寄存器的默认值为 **0x68**。

位 **0** 和 **7** 保留。（硬编码为 **0**）。

参数：

WHO_AM_I 包含 **MPU-60X0** 的 6 位 **I2C** 地址。
 上电复位的值从位 **6** 到位 **1** 为：**110100**。
 The Power-On-Reset value of Bit6:Bit1 is 110 100