

1.MPU6050 是什么？

MPU6050 是一个 6 轴运动处理组件，包含了 3 轴加速度 和 3 轴陀螺仪。

MPU-6000 为全球首例整合性 6 轴运动处理组件，相较于多组件方案，免除了组合陀螺仪与加速器时之轴间差的问题，减少了大量的包装空间。MPU-6000 整合了 3 轴陀螺仪、3 轴加速器，并含可藉由第二个 I2C 端口连接其他厂牌之加速器、磁力传感器、或其他传感器的数位运动处理(DMP: Digital Motion Processor)硬件加速引擎，由主要 I2C 端口以单一数据流的形式，向应用端输出完整的 9 轴融合演算技术

InvenSense 的运动处理资料库，可处理运动感测的复杂数据，降低了运动处理运算对操作系统的负荷，并为应用开发提供架构化的 API。

MPU-6000 的角速度全格感测范围为 ± 250 、 ± 500 、 ± 1000 与 $\pm 2000^\circ/\text{sec}$ (dps)，可准确追縱快速与慢速动作，并且，用户可程式控制的加速器全格感测范围为 $\pm 2g$ 、 $\pm 4g$ $\pm 8g$ 与 $\pm 16g$ 。产品传输可透过最高至 400kHz 的 I2C 或最高达 20MHz 的 SPI。

MPU-6000 可在不同电压下工作，VDD 供电电压介为 $2.5V \pm 5\%$ 、 $3.0V \pm 5\%$ 或 $3.3V \pm 5\%$ ，逻辑接口 VDDIO 供电为 $1.8V \pm 5\%$ 。MPU-6000 的包装尺寸 $4 \times 4 \times 0.9\text{mm}$ (QFN)，在业界是革命性的尺寸。其他的特征包含内建的温度感测器、包含在运作环境中仅有 $\pm 1\%$ 变动的振荡器。

2.加速度传感器是干嘛用的？

总而言之，加速度传感器，其实是力传感器。用来检查上下左右前后哪几个面都受了多少力（包括重力），然后计算**角度**。

3.陀螺仪是干嘛用的？

简而言之，陀螺仪就是**角速度**检测仪。比如，一块板，以 X 轴为轴心，在一秒钟的时间转到了 90 度，那么它在 X 轴上的角速度就是 90 度/秒

4.MPU6050 分辨率是多少？

3 轴加速度 和 3 轴陀螺仪分别用了 3 个 16 位的 ADC, 也就是说，加速度有 3 个 16 位 ADC，其中每个轴使用了一个。也是说，每个轴输出的数据，是 2^{16} 也就是 $-32768 \sim +32768$ 。陀螺仪也是一样。

5. 单位换算

上面说的 $-32768 \sim +32768$ ，那么这个数字到底代表了什么呢？比如陀螺仪 32768 到底是指角速度达到多少度/秒？

这个其实是根据 MPU6050 设置的量程来决定的，量程不一样，32768 代表的值就不一样。

MPU6050 的量程设置，在 MPU6050::initialize()（MPU6050.cpp 库）初始化函数中进行了设置：
`setFullScaleGyroRange(MPU6050_GYRO_FS_250);`
`setFullScaleAccelRange(MPU6050_ACCEL_FS_2);`
分别设置为，250 度/秒，2g

按陀螺仪来说，MPU6050 有四个量程可选：

±250, ±500, ±1000, ±2000 度/s
比方说, 设置了是 ±250 ,那么-32768

STM32 自带 I2C, 但一般有两个 I2C1 (PB6,PB7) 和 I2C2 (PB10,PB11), 而且, I2C 分为硬件、和模拟。

软件 i2c 是程序员使用程序控制 SCL,SDA 线输出高低电平, 模拟 i2c 协议的时序.

硬件 i2c 程序员只要调用 i2c 的控制函数即可, 不用直接的去控制 SCL,SDA 高低电平的输出

本模块采用的是 IIC 通信方式, 所以我们只需要连接四跟线就可以完成电路的连接, 简单方便!

原始数据有: AX、AY、AZ

简单的算法之后可以得到 Roll, pitch, yaw

参考 MPU-6050 数据手册

引脚说明:

引脚编号	MPU-6000	MPU-6050	引脚名称	描述
1	Y	Y	CLKIN	可选的外部时钟输入，如果不用则连到 GND
6	Y	Y	AUX_DA	I2C 主串行数据，用于外接传感器
7	Y	Y	AUX_CL	I2C 主串行时钟，用于外接传感器
8	Y		/CS	SPI 片选 (0=SPI mode)
8		Y	VLOGIC	数字 I/O 供电电压
9	Y		AD0/SDO	I2C Slave 地址 LSB (AD0); SPI 串行数据输出 (SDO)
9		Y	AD0	I2C Slave 地址 LSB (AD0)
10	Y	Y	REGOUT	校准滤波电容连线
11	Y	Y	FSYNC	帧同步数字输入
12	Y	Y	INT	中断数字输出 (推挽或开漏)
13	Y	Y	VDD	电源电压及数字 I/O 供电电压
18	Y	Y	GND	电源地
19, 21, 22	Y	Y	RESV	预留，不接
20	Y	Y	CPOUT	电荷泵电容连线
23	Y		SCL/SCLK	I2C 串行时钟 (SCL); SPI 串行时钟 (SCLK)
23		Y	SCL	I2C 串行时钟 (SCL)
24	Y		SDA/SDI	I2C 串行数据 (SDA); SPI 串行数据输入 (SDI)
24		Y	SDA	I2C 串行数据 (SDA)
2, 3, 4, 5, 14, 15, 16, 17	Y	Y	NC	不接

VDD 供电电压为 2.5V±5%、3.0V±5%、3.3V±5%；VDDIO 为 1.8V±5%

内建振荡器在工作温度范围内仅有±1%频率变化。可选外部时钟输入 32.768kHz 或 19.2MHz

找出几个重要的寄存器：

1) Register 25 – 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]							

1) SMPLRT_DIV

该寄存器指定陀螺仪输出率的分频，用来产生 MPU-60X0 的采样率。

传感器寄存器的输出、FIFO 输出、DMP 采样和运动检测的都是基于该采样率。

采样率的计算公式

采样率 =

当数字低通滤波器没有使能的时候，陀螺仪的输出平路等于 8KHZ，反之等于 1KHZ。

2) Register 26 – 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]		

数字低通滤波器

陀螺仪和加速度计配置外部帧同步（FSYNC）引脚采样和数字低通滤波器（DLPF）。通过配置 EXT_SYNC_SET，可以对连接到 FSYNC 引脚的一个外部信号进行采样。

锁存器将复位到当前的 FSYNC 信号状态。

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]

数字低通滤波器是由 DLPF_CFG 来配置，根据下表中 DLPF_CFG 的值对加速度传感器和陀螺仪滤波

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

3) Register 27 – 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]		-	-	-

- 1) XG_ST 设置此位，X 轴陀螺仪进行自我测试。
- 2) YG_ST 设置此位，Y 轴陀螺仪进行自我测试。
- 3) ZG_ST 设置此位，Z 轴陀螺仪进行自我测试。
- 4) FS_SEL 2 位无符号值。选择陀螺仪的量程。

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测试陀螺仪的机械和电气部分，通过设置该寄存器的 XG_ST、YG_ST 和 ZG_ST bits 可以激活陀螺仪对应轴的自检。每个轴的检测可以独立进行或同时进行。

自检的响应 = 打开自检功能时的传感器输出 - 未启用自检功能时传感器的输出

在 MPU-6000/MPU-6050 数据手册的电气特性表中已经给出了每个轴的限制范围。当自检的响应值在规定的范围内，就能够通过自检；反之，就不能通过自检。

根据下表，FS_SEL 选择陀螺仪输出的量程：

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

4) Register 28 – 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]		-		

- 1) XA_ST 设置为 1 时，X 轴加速度感应器进行自检。
- 2) YA_ST 设置为 1 时，Y 轴加速度感应器进行自检。
- 3) ZA_ST 设置为 1 时，Z 轴加速度感应器进行自检。
- 4) AFS_SEL 2 位无符号值。选择加速度计的量程。

具体细节和上面陀螺仪的相似。

根据下表，AFS_SEL 选择加速度传感器输出的量程。

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

5) Registers 59 to 64 – Accelerometer Measurements (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]							

- 1) ACCEL_XOUT 16 位 2' s 补码值。
存储最近的 X 轴加速度感应器的测量值。
- 2) ACCEL_YOUT 16 位 2' s 补码值。
存储最近的 Y 轴加速度感应器的测量值。
- 3) ACCEL_ZOUT 16 位 2' s 补码值。
存储最近的 Z 轴加速度感应器的测量值。

这些寄存器存储加速感应器最近的测量值。

加速度传感器寄存器，连同温度传感器寄存器、陀螺仪传感器寄存器和外部感应数据寄存器，都由两部分寄存器组成（类似于 STM32F10X 系列中的影子寄存器）：一个内部寄存器，用户不可见。另一个用户可读的寄存器。内部寄存器中数据在采样的时候及时的到更新，仅在串行通信接口不忙碌时，才将内部寄存器中的值复制到用户可读的寄存器中去，避免了直接对感应测量值的突发访问。

在寄存器 28 中定义了每个 16 位的加速度测量值的最大范围，对于设置的每个最大范围，都对应一个加速度的灵敏度 ACCEL_xOUT，如下面的表中所示：

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

6) Registers 65 and 66 – Temperature Measurement (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]							

- 1) TEMP_OUT 16 位有符号值。
存储的最近温度传感器的测量值。

7) Registers 67 to 72 – Gyroscope Measurements (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]							

这个和加速度感应器的寄存器相似

对应的灵敏度：

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/ $^{\circ}/\text{s}$

8) Register 107 – 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]		

该寄存器允许用户配置电源模式和时钟源。它还提供了一个复位整个器件的位，和一个关闭温度传感器的位

- 1) DEVICE_RESET 置 1 后所有的寄存器复位，随后 DEVICE_RESET 自动置 0.
- 2) SLEEP 置 1 后进入睡眠模式
- 3) CYCLE 当 CYCLE 被设置为 1，且 SLEEP 没有设置，MPU-60X0 进入循环模式，为了从速度传感器中获得采样值，在睡眠模式和正常数据采集模式之间切换，每次获得一个采样数据。在 LP_WAKE_CTRL (108) 寄存器中，可以设置唤醒后的采样率和被唤醒的频率。
- 4) TEMP_DIS 置 1 后关闭温度传感器
- 5) CLKSEL 指定设备的时钟源

时钟源的选择：

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

9) Register 117 – Who Am I (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]						-

上电复位的第 6 位到第 1 位值为：110100

为了让两个 MPU-6050 能够连接在一个 I2C 总线上，当 AD0 引脚逻辑低电平时，设备的地址是

(2013.01.24)



MPU-6000 可以使用 SPI 和 I2C 接口，而 MPU-6050 只能使用 I2C，其中 I2C 的地址由 AD0 引脚决定;寄存器共 117 个，挺多的，下面的是精简常用的，根据具体的要求，适当的添加。


```
#define //采样率分频，典型值：0x07(125Hz) */
#define // 低通滤波频率，典型值：0x06(5Hz) */
#define // 陀螺仪自检及测量范围，典型值：0x18(不自检，2000deg/s) */
#define

#define // 存储最近的 X 轴、Y 轴、Z 轴加速度感应器的测量值 */
#define
#define
#define
#define
#define

#define // 存储的最近温度传感器的测量值 */
#define

#define
#define
#define
#define
#define
#define

#define
#define
```

编程时用到的关于 I2C 协议规范：

信号	描述
S	开始标志：SCL 为高时 SDA 的下降沿
AD	从设备地址（Slave 地址）
W	写数据位（0）
R	读数据位（1）
ACK	应答信号：在第 9 个时钟周期 SCL 为高时，SDA 为低
NACK	拒绝应答：在第 9 个时钟周期，SDA 一直为高
RA	MPU-60X0 内部寄存器地址
DATA	发送或接受的数据
P	停止标志：SCL 为高时 SDA 的上升沿

