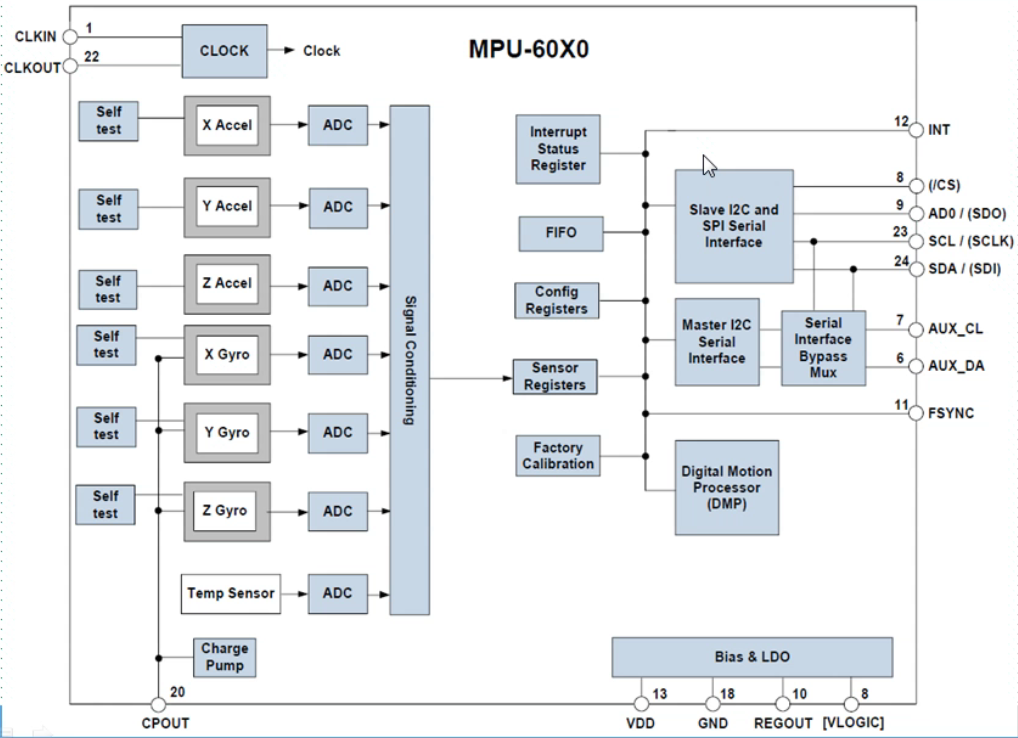
开始搞事搞事搞事！

缩实话MPU60X0这玩意听过好多好多次啦，到现在腾出手来搞搞。

60X0大致介绍：

MPU-60X0是9轴运动处理传感器，集成了3轴MEMS陀螺仪，3轴MEMS加速度计，可用I2C接口连接一个第三方的数字传感器,比如磁力计，也可连接非惯性的数字传感器,比如压力传感器，以及一个可扩展的**数字运动处理器DMP**，扩展后就可以通过其I2C或SPI接口输出一个9轴信号（SPI接口仅在6000可用）。DMP从陀螺仪、加速度计以及外接的传感器接收并处理数据，处理结果可以从DMP寄存器读出，或通过FIFO缓冲。DMP有权使用MPU的一个外部引脚产生中断。



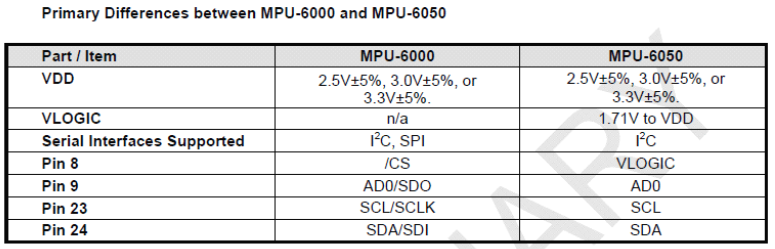
MPU-60X0包含一个1024字节的FIFO寄存器，有助于降低系统功耗，FIFO配置寄存器可以决定什么让什么数据进入，可选陀螺仪、加速度计、温度、外部传感器，以及FSYNC输入。一个FIFO计数器可以跟踪存入FIFO的字节数。

MPU-60X0对陀螺仪(角速度感测器)和加速度计分别用了三个16位的ADC,将其测量的模拟量转化为可输出的数字量。为了精确跟踪快速和慢速的运动，传感器的测量范围都是用户可控的，陀螺仪可测范围为±250，±500，±1000，±2000°/秒（dps），运作电流5mA，待命电流5uA；加速度计可测范围为±2，±4，±8，±16g，运做电流500uA，省电模式电流40uA。

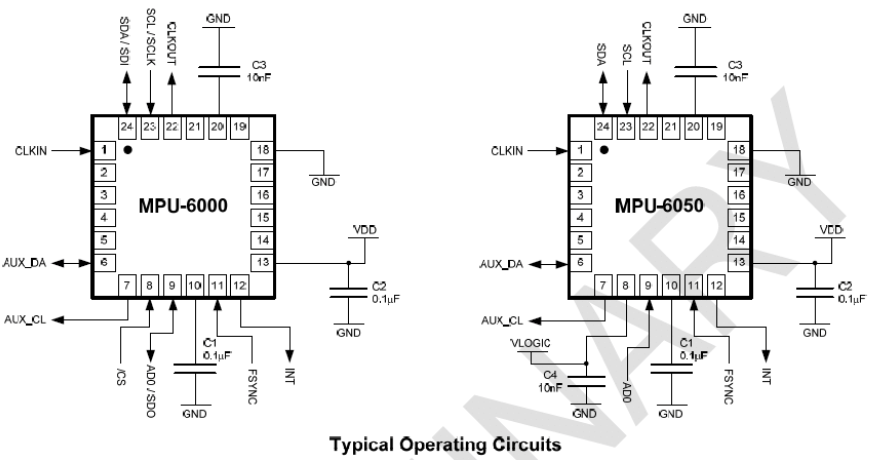
和所有设备寄存器之间的通信采用400kHZ的I2C接口或1MHZ的SPI接口（SPI仅6000可用），对于需要高速传输的应用，对寄存器的读取和中断可用20MHZ的SPI。另外片上还内嵌了一个温度传感器和在工作温度环境下仅有±1%频率变化的振荡器，可选32.768kHZ或19.2MHZ的外部时钟输入。

芯片尺寸4\*4\*0.9mm，采用QFN封装（无引线方形封装），可承受最大10000g的冲击，有可编程的低通滤波器。

6000与6050区别：

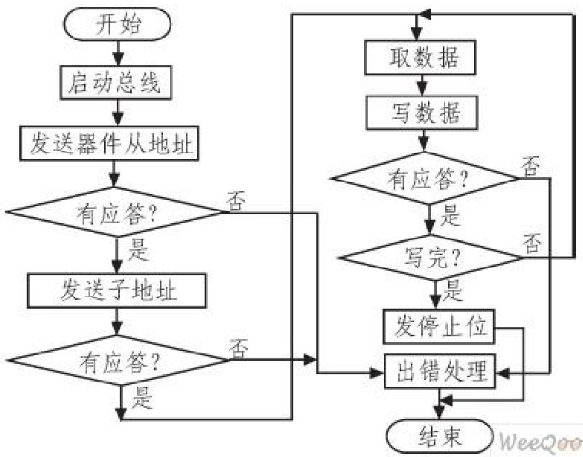


VLOGIC引脚用来为I2C输出提供逻辑电平





程序流程鬼：



**寄存器：**



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 b1101000 ，当AD0引脚逻辑高电平时，设备的地址是 b1101001。

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。

**1.Power Management 1（PWR\_MGMT\_1）// 电源管理寄存器1**

**#define PWR\_MGMT\_1  0x6B // 电源管理，典型值：0x00(正常启用) \*/**



写0X00给寄存器PWM\_MGMT\_1完成传感器的启动(为固定模式)。

Bit7---DEVICE\_RESE，复位MPU050，复位完成后，自动清零。

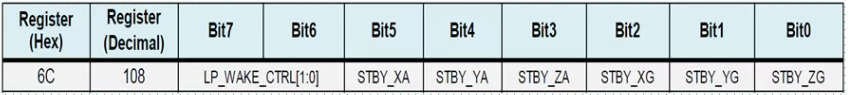
Bit6---SLEEP=1，进入睡眠模式；SLEEEP=0，正常工作模式。

Bit3---TEMP\_DIS，用于设置是否使能温度传感器，设置为0，则使能。

CLKSEL[2：0],用于选择系统时钟源：



1. **Power Management 2（PWR\_MGMT\_2）// 电源管理寄存器2**

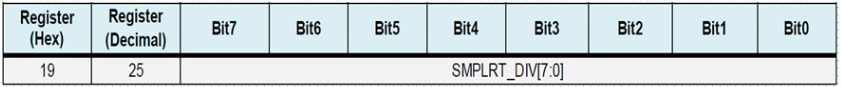


Bit7---LP\_WAKE\_CTRL用于控制低功耗时的唤醒频率。

剩下的6位，分别控制加速度计和陀螺仪的x/y/z轴是否进入待机模式，0为不进入待机模式。

**3.Sample Rate Divider（SMPRT\_DIV）//陀螺仪采样率分频寄存器**

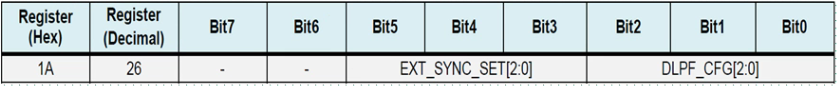
**#define SMPLRT\_DIV  0x19    //设置陀螺仪采样频率，典型值：0x07(125Hz) \*/**

  
 陀螺仪采样频率 = 陀螺仪的输出速度 / (1 + SMPLRT\_DIV)。

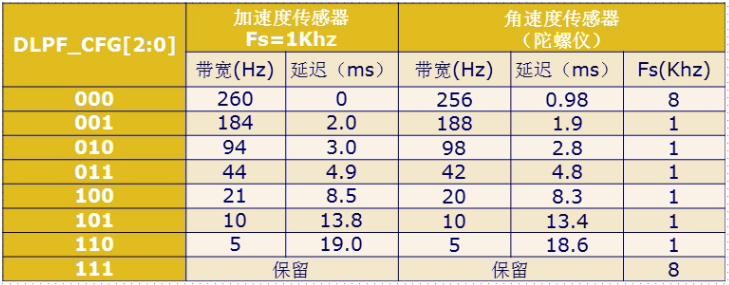
这里陀螺仪的输出频率，是1KHZ或者8KHZ，与数字低通滤波器(DLPF)的设置有关，当DLPF\_CFG=0或7（没有使能）的时候，频率为8khz,其他情况是1KHZ。而且DLPF滤波频率一般设置为采样率的一半。采样率，假定设置为50hz，那么：SMPLRT\_DIV=1000/50-1=19。

**4.Configuration(CONFIG)//配置寄存器**

**#define CONFIG    0x1A   // 设置低通滤波器频率，典型值：0x06(5Hz) \*/**



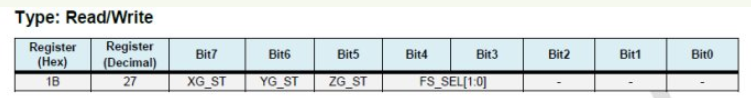




带宽=1/2采样率

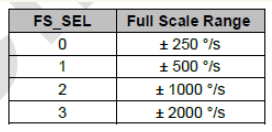
**5.Gyroscope Configuration(GYRO\_CONFIG)//陀螺仪配置寄存器**

**#define GYRO\_CONFIG  0x1B   //设置陀螺仪自检及测量范围，典型值：0x18(不自检，2000deg/s) \*/**



XG\_ST、YG\_ST、ZG\_ST为0时，陀螺仪所在轴上不进行自检测，为1时进行自检测。

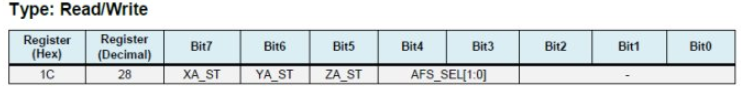
陀螺仪测量范围：



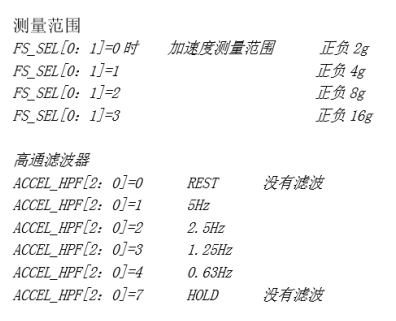
一般设置为3，因为陀螺仪的ADC为16位分辨率，所以得到灵敏度为：65536/4000 = 16.4LSB/ (°/S)

**6.Accelerometer Configuration（ACCEL\_CONFIG）//加速度传感器配置寄存器**

**#define ACCEL\_CONFIG 0x1C   // 设置加速计自检测、测量范围及高通滤波器频率，典型值：0x01(不自检，2G，5Hz) \*/**



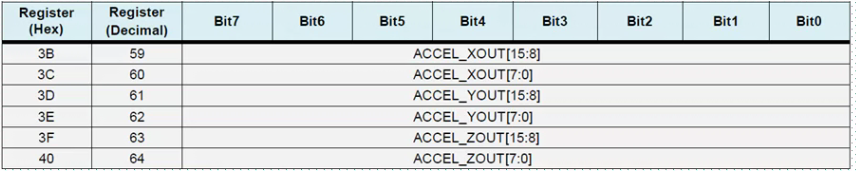
XA\_ST、YA\_ST、ZA\_ST为0时加速度所在轴上不进行自检测，为1时进行自检测。



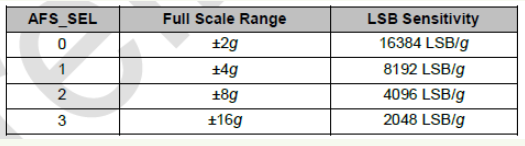
一般设置为0，即±2g，因为加速度计传感器的ADC也是16位，所以得到灵敏度为：65536/4=16384LSB/g。

**7.Accelerometer Measurements（ACCEL\_XOUT\_H, ACCEL\_XOUT\_L, ACCEL\_YOUT\_H, ACCEL\_YOUT\_L, ACCEL\_ZOUT\_H, and ACCEL\_ZOUT\_L）//加速度传感器数据输出寄存器**

**#define ACCEL\_XOUT\_H 0x3B  // 存放X轴加速度的高八位 \*/  
#define ACCEL\_XOUT\_L 0x3C // 存放X轴加速度的低八位 \*/  
#define ACCEL\_YOUT\_H 0x3D // 存放Y轴加速度的高八位 \*/  
#define ACCEL\_YOUT\_L 0x3E // 存放Y轴加速度的低八位 \*/  
#define ACCEL\_ZOUT\_H 0x3F // 存放Z轴加速度的高八位 \*/  
#define ACCEL\_ZOUT\_L 0x40 // 存放Z轴加速度的低八位 \*/**

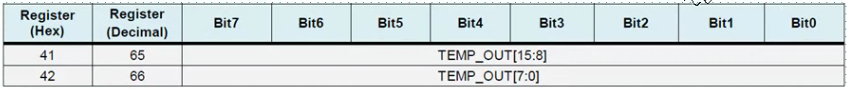


在寄存器28中定义了每个16位的加速度测量值的最大范围，对于设置的每个最大范围，都对应一个加速度的灵敏度ACCEL\_xOUT，如下面的表中所示：



**8.Temperature Measurement （TEMP\_OUT\_H and TEMP\_OUT\_L）//温度传感器数据输出寄存器**

**#define TEMP\_OUT\_H  0x41   // 存放温度高八位 \*/  
#define TEMP\_OUT\_L  0x42 // 存放温度低八位 \*/**



温度换算公式：Temperature = 36.53 + regval/340

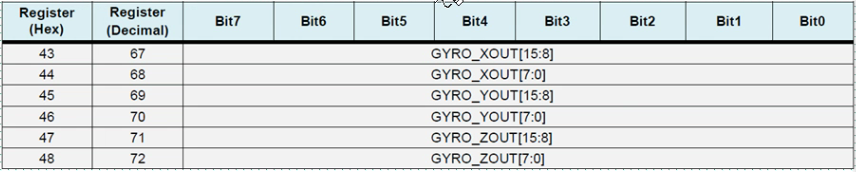
其中，Temperature为计算得到的温度值，单位为°C，regval为从0x41和0x42读到的温度传感器值。

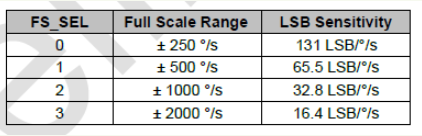
**9.Gyroscope Measurements（GYRO\_XOUT\_H, GYRO\_XOUT\_L, GYRO\_YOUT\_H, GYRO\_YOUT\_L,GYRO\_ZOUT\_H,andGYRO\_ZOUT\_L)//陀螺仪数据输出寄存器**

**#define GYRO\_XOUT\_H  0x43 // 存放X轴角速度的高八位 \*/  
#define GYRO\_XOUT\_L  0x44 // 存放X轴角速度的低八位 \*/  
#define GYRO\_YOUT\_H  0x45 // 存放Y轴角速度的高八位 \*/**

**#define GYRO\_YOUT\_L  0x46 // 存放Y轴角速度的低八位 \*/  
#define GYRO\_ZOUT\_H  0x47 // 存放Z轴角速度的高八位 \*/  
#define GYRO\_ZOUT\_L  0x48 // 存放Z轴角速度的低八位 \*/**

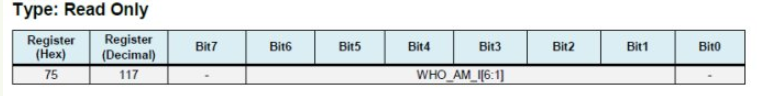
这个和加速度感应器的寄存器相似，对应的灵敏度：





**10.Register 117--Who Am I （WHO\_AM\_I）**

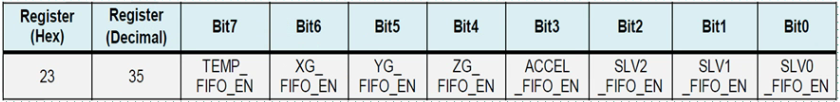
**#define WHO\_AM\_I  0x75 //IIC地址寄存器(默认数值0x68，只读) \*/**



WHO\_AM\_I中的内容是MPU-60X0的6位I2C地址

上电复位的第6位到第1位值为：110100

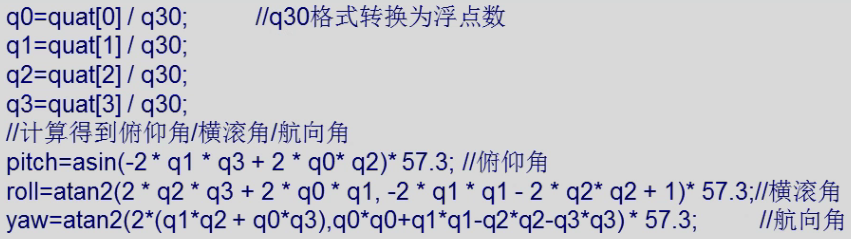
**11.FIFO使能寄存器 0x23**



该寄存器用于控制FIFO使能，在简单读取传感器数据的时候，可以不用FIFO,设置对应为0，即可禁止；设置为1，则使能FIFO。加速度传感器的3个轴，全由Bit3---ACCEL\_FIFO\_EN控制，只有该位置1，则加速度传感器的三个通道都开启FIFO了。

**姿态角（欧拉角）：**

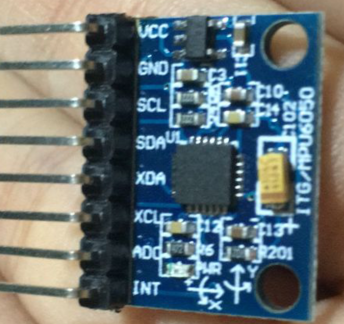
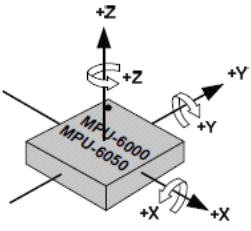
MOU6050 DMP输出的是姿态解算后的四元数，采用q30格式，也就是放大的2的30次方，要得到欧拉角，就得做一个转换。

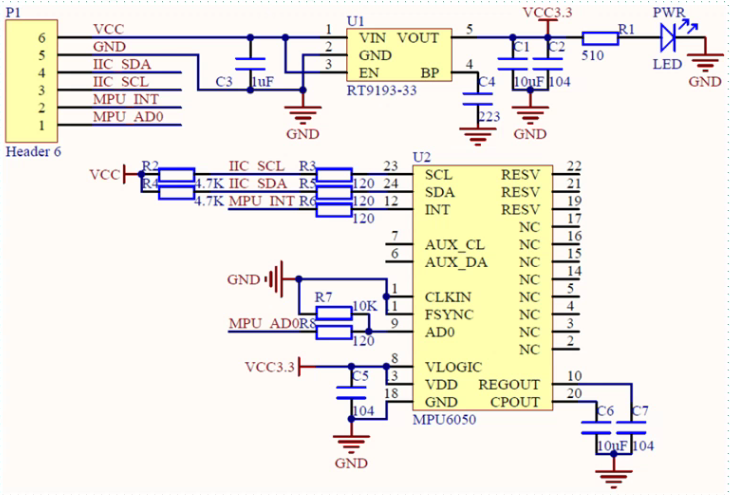


quat[0]~quat[3]:是MPU6050的DMP解算后的四元数，q30格式。

q30：是一个常量1073741824，即2的30次方。

57.3：是弧度转换为角度，即180/π，这样的结果就是以度为单位的。





ATK-MPU6050模块原理图