

MPU6050运动中断应用

MPU6050运动中断相关寄存器

中断使能寄存器: INT_ENABLE

FIFO控制寄存器: USER_CTRL

FIFO使能寄存器: FIFO_EN

中断引脚配置寄存器: INT_PIN_CFG

运动状态寄存器: MOT_DETECT_STATUS

自由落体中断阈值寄存器: FF_THR

自由落体中断时间寄存器: FF_DUR

加速度中断阈值寄存器: MOT_THR

加速度中断时间寄存器: MOT_DUR

静止中断阈值寄存器: ZRMOT_THR

静止中断时间寄存器: ZRMOT_DUR

MPU6050初始化参考例程

MPU6050运动中断初始化参考例程

自由落体中断参考例程

加速度中断参考例程

静止中断参考例程

MPU6050运动中断相关寄存器

由于这样或那样的原因，MPU6050的技术文档中省略了很多寄存器，而这些被刻意省略的寄存器往往涉及到各类高级应用，如较为常用的运动中断（自由落体中断、加速度中断和静止中断）、内部DMP数据融合和内部FIFO等。在此感谢运用反编译等技术手段整理出相关寄存器和DMP驱动的大神。

除基本寄存器（PWR_MGMT、CONFIG、ACCEL_CONFIG等）之外，运动中断应用所涉及到的相关内部寄存器主要包括：

1. **中断使能寄存器: INT_ENABLE**，寄存器地址0x38，为MPU6050所有中断输出的开关寄存器，用于使能运动中断、FIFO溢出中断和数据中断等；

- 2. **FIFO控制寄存器：USER_CTRL**，寄存器地址0x6A，该寄存器用于使能FIFO，并可控制MPU6050的I2C主机。应用运动中中断功能时，应关闭FIFO和I2C主机；
- 3. **FIFO使能寄存器：FIFO_EN**，寄存器地址0x23，该寄存器用于使能各个FIFO功能，在应用运动中中断功能时应关闭；
- 4. **中断引脚配置寄存器：INT_PIN_CFG**，寄存器地址0x37，该寄存器用于设置INT中断引脚的电平标准和驱动方式（推挽、开漏）等；
- 5. **运动状态寄存器：MOT_DETECT_STATUS**，寄存器地址0x61，该寄存器用于在触发运动中中断（自由落体中断、加速度中断和静止中断）时标示中断的类型。
- 6. **自由落体中断阈值寄存器：FF_THR**，寄存器地址0x1D，当使能自由落体中断时，该寄存器的值决定中断的触发阈值，数值越高，对自由落体的检测越不灵敏；
- 7. **自由落体中断时间寄存器：FF_DUR**，寄存器地址0x1E，当使能自由落体中断时，该寄存器的值决定中断的持续时间阈值，当自由落体的持续时间达到设定阈值时触发中断。数值越高，对自由落体检测的时间阈值越长；
- 8. **加速度中断阈值寄存器：MOT_THR**，寄存器地址0x1F，当使能加速度中断时，该寄存器的值决定中断的触发阈值，数值越高，触发中断所需的加速度越大。
- 9. **加速度中断时间寄存器：MOT_DUR**，寄存器地址0x20，当使能加速度中断时，该寄存器的值决定中断的持续时间阈值，当加速度值持续时间达到设定阈值时触发中断。数值越高，触发中断所需的加速度持续时间越长；
- 10. **静止中断阈值寄存器：ZRMOT_THR**，寄存器地址0x21，当使能静止中断时，若当前加速度计输出的三轴值均小于静止中断阈值，则产生静止中断。数值越高，触发静止中断的要求越宽松。
- 11. **静止中断时间寄存器：ZRMOT_DUR**，寄存器地址0x22，当使能静止中断时，该寄存器的值决定触发静止中断所需的持续时间，数值越高，触发中断所需的静止持续时间越长。

除特殊说明外，各个寄存器均为逻辑1使能

中断使能寄存器：INT_ENABLE

寄存器名	助记符	寄存器地址
中断使能寄存器	INT_ENABLE	0x38

寄存器位	功能
bit7	自由落体中断使能
bit6	加速度中断使能

寄存器位	功能
bit5	静止中断使能
bit4	运动检测中断使能
bit3	FIFO溢出中断使能
bit2	无定义
bit1	无定义
bit0	数据就绪中断使能

FIFO控制寄存器：USER_CTRL

寄存器名	助记符	寄存器地址
FIFO控制寄存器	USER_CTRL	0x6A

寄存器位	功能
bit7	使能内部DMP
bit6	使能内部FIFO
bit5	使能辅助I2C总线主机
bit4	(仅MPU6000) 转换为SPI模式
bit3	复位内部DMP
bit2	复位内部FIFO，此时FIFO_EN清零
bit1	复位辅助I2C总线主机
bit0	复位所有传感器寄存器与外设寄存器（软重启）

FIFO使能寄存器：FIFO_EN

寄存器名	助记符	寄存器地址
FIFO使能寄存器	FIFO_EN	0x23

寄存器位	功能
bit7	使能温度FIFO
bit6	使能陀螺仪X轴FIFO
bit5	使能陀螺仪Y轴FIFO
bit4	使能陀螺仪Z轴FIFO
bit3	使能加速度计FIFO
bit2	使能辅助I2C总线从设备2数据FIFO
bit1	使能辅助I2C总线从设备1数据FIFO
bit0	使能辅助I2C总线从设备0数据FIFO

中断引脚配置寄存器：INT_PIN_CFG

寄存器名	助记符	寄存器地址
中断引脚配置寄存器	INT_PIN_CFG	0x37

寄存器位	功能
bit7	配置INT引脚逻辑电平，当设置为0时，INT引脚静息为低电平，触发中断时产生高电平信号
bit6	配置引脚驱动，0：推挽模式；1：开漏模式
bit5	配置中断输出方式，0：50us电平脉冲；1：持续信号直到中断结束
bit4	配置中断锁存清除方式，0：只读（不允许手动清零）；1：可读写
bit3	配置FSYNC中断逻辑电平，0：高电平；1：低电平
bit2	FSYNC端口中断使能，0：禁用；1：启用
bit1	I2C辅助总线状态标志位，当此位为1时，主机可以直接访问I2C支路从机
bit0	CLKOUT端口输出使能，0：禁用；1：输出参考时钟

运动状态寄存器：MOT_DETECT_STATUS

寄存器名	助记符	寄存器地址
运动状态寄存器	MOT_DETECT_STATUS	0x61

寄存器位	功能
bit7	X轴反向运动检测中断状态标志位
bit6	X轴正向运动检测中断状态标志位
bit5	Y轴反向运动检测中断状态标志位
bit4	Y轴正向运动检测中断状态标志位
bit3	Z轴反向运动检测中断状态标志位
bit2	Z轴正向运动检测中断状态标志位
bit1	无定义
bit0	静止中断状态标志位

自由落体中断阈值寄存器：FF_THR

寄存器名	助记符	寄存器地址
自由落体中断阈值寄存器	FF_THR	0x1D

寄存器位	功能
bit0-7	该寄存器存储自由落体中断的加速度检测阈值，单位为1LSB = 2mg。加速度计的任意一个轴的数据超过该阈值时就会触发自由落体中断计时，若超过阈值的加速度持续时间超过自由落体中断计时时间，就会产生自由落体中断

自由落体中断时间寄存器：FF_DUR

寄存器名	助记符	寄存器地址
自由落体中断时间寄存器	FF_DUR	0x1E

寄存器位	功能
bit0-7	该寄存器存储自由落体中断的时间阈值，单位为ms

加速度中断阈值寄存器：MOT_THR

寄存器名	助记符	寄存器地址
加速度中断阈值寄存器	MOT_THR	0x1F

寄存器位	功能
bit0-7	该寄存器存储加速度中断的加速度检测阈值，单位为1LSB = 2mg。加速度计的任意一个轴的数据超过该阈值时就会触发加速度中断计时，若超过阈值的加速度持续时间超过加速度中断计时时间，就会产生加速度中断

加速度中断时间寄存器：MOT_DUR

寄存器名	助记符	寄存器地址
加速度中断时间寄存器	MOT_DUR	0x20

寄存器位	功能
bit0-7	该寄存器存储加速度中断的时间阈值，单位为ms

静止中断阈值寄存器：ZRMOT_THR

寄存器名	助记符	寄存器地址
静止中断阈值寄存器	ZRMOT_THR	0x21

寄存器位	功能
bit0-7	该寄存器存储静止中断的加速度检测阈值，单位为1LSB = 2mg。加速度计的三个轴的数据均小于该阈值时就会触发静止中断计时，若静止中断计时结束后加速度计值均未超过阈值，就会产生静止中断

静止中断时间寄存器：ZRMOT_DUR

寄存器名	助记符	寄存器地址
静止中断时间寄存器	ZRMOT_DUR	0x22

寄存器位	功能
bit0-7	该寄存器存储静止中断的时间阈值，单位为ms

MPU6050初始化参考例程

```
1  uchar Init_MPU6050(void)           //MPU6050初始化
2  {
3      uchar ack=0;
4      I2C_Write(GYRO_ADDRESS,PWR_MGMT_1, 0x00);    //解除休眠状态
5      I2C_Write(GYRO_ADDRESS,SMPLRT_DIV, 0x07);    //陀螺仪采样率,典型值:0x07(125Hz)
6      I2C_Write(GYRO_ADDRESS,CONFIG, 0x06);        //低通滤波频率,典型值:0x06(5Hz)
7      I2C_Write(GYRO_ADDRESS,GYRO_CONFIG, 0x18);   //陀螺仪自检及测量范围,典型值: 0x18(不自检)
8      I2C_Write(GYRO_ADDRESS,ACCEL_CONFIG, 0x01);  //加速计自检、测量范围及高通滤波频率,典型值:0x01(±2g)
9      I2C_Write(GYRO_ADDRESS,INT_ENABLE,0X00);     //关闭所有中断
10     I2C_Write(GYRO_ADDRESS,USER_CTRL,0X00);      //I2C主模式关闭
11     I2C_Write(GYRO_ADDRESS,FIFO_EN,0X00);        //关闭FIFO
12     I2C_Write(GYRO_ADDRESS,INT_PIN_CFG,0X80);     //中断的逻辑电平模式,设置为0, 中断信号引脚为推挽IO
13     ack=I2C_Read(GYRO_ADDRESS,WHO_AM_I);         //查询IMU是否在线
14     if(ack==MPU_ADDR)
15     {
16         I2C_Write(GYRO_ADDRESS,PWR_MGMT_1,0X02); //设置CLKSEL,PLL X 轴为参考
17         I2C_Write(GYRO_ADDRESS,PWR_MGMT_2,0X00); //加速度陀螺仪都工作
18         MPU_Set_Rate(200);                        //设置采样率为200-500Hz
19     }else return 1;
20     return 0;
21 }
```

MPU6050运动中断初始化参考例程

自由落体中断参考例程

```
1  void MPU_Fall_Init(void)           //自由落体中断初始化
2  {
3      I2C_Write(GYRO_ADDRESS,FF_THR,0x14);         //设置自由落体加速度阈值为20mg
4      I2C_Write(GYRO_ADDRESS,FF_DUR,0x0A);         //设置自由落体检测时间10ms
5      I2C_Write(GYRO_ADDRESS,CONFIG,0x04);         //配置外部引脚采样和DLPF数字低通滤波器
```

```

6 | I2C_Write(GYRO_ADDRESS,ACCEL_CONFIG,0x1C); //加速度传感器量程和高通滤波器配置
7 | I2C_Write(GYRO_ADDRESS,INT_PIN_CFG,0X1C); //INT引脚低电平平时
8 | I2C_Write(GYRO_ADDRESS,INT_ENABLE,0x40); //中断使能寄存器
9 | }

```

加速度中断参考例程

```

1 | void MPU_Motion_Init(void) //加速度中断初始化
2 | {
3 |     I2C_Write(GYRO_ADDRESS,MOT_THR,0x25); //设置加速度阈值为74mg
4 |     I2C_Write(GYRO_ADDRESS,MOT_DUR,0x14); //设置加速度检测时间20ms
5 |     I2C_Write(GYRO_ADDRESS,CONFIG,0x04); //配置外部引脚采样和DLPF数字低通滤波器
6 |     I2C_Write(GYRO_ADDRESS,ACCEL_CONFIG,0x1C); //加速度传感器量程和高通滤波器配置
7 |     I2C_Write(GYRO_ADDRESS,INT_PIN_CFG,0X1C); //INT引脚低电平平时
8 |     I2C_Write(GYRO_ADDRESS,INT_ENABLE,0x40); //中断使能寄存器
9 | }

```

静止中断参考例程

```

1 | void MPU_Zero_Motion_Init(void) //静止中断初始化
2 | {
3 |     I2C_Write(GYRO_ADDRESS,ZRMOT_THR,0x20); //设置加速度阈值为64mg
4 |     I2C_Write(GYRO_ADDRESS,ZRMOT_DUR,0x20); //设置静止检测时间32ms
5 |     I2C_Write(GYRO_ADDRESS,CONFIG,0x04); //配置外部引脚采样和DLPF数字低通滤波器
6 |     I2C_Write(GYRO_ADDRESS,ACCEL_CONFIG,0x1C); //加速度传感器量程和高通滤波器配置
7 |     I2C_Write(GYRO_ADDRESS,INT_PIN_CFG,0X1C); //INT引脚低电平平时
8 |     I2C_Write(GYRO_ADDRESS,INT_ENABLE,0x40); //中断使能寄存器
9 | }

```