

基于 STM32 的硬件 I2C 读取 MPU6050 数据 - 单片机干货 - 中国电子技术论坛 - 最好最受欢迎电子论坛!

笔记本: STM32学习剪辑资料
创建时间: 2017/9/2 16:35
URL: http://bbs.elecfans.com/jishu_485014_1_1.html

[单片机经验] 基于 STM32 的硬件 I2C 读取 MPU6050 数据

 [复制链接]



tyw0403

实习生

51单片机
开发板

2000分钟
教学视频
100个
实例代码

马上购买

发表于 2015-5-27 17:43:29 24165 查看 212 回复

只看该作者

本帖最后由 tyw0403 于 2015-6-4 15:06 编辑

根据网上一些资料，调试了两天，终于读取到了MPU6050的六个数据，调试过程当中也遇到了很多问题，下面在此总结分享一下。

MPU6050其实就是一个 I2C 器件，里面有很多寄存器（但是我们用到的只有几个），我们通过读写寄存器来操作这个芯片。所以首要问题就是 STM32 和 MPU6050 的 I2C 通信。

1、配置 STM32 （用I2C1：PB6——SCL； PB7——SDA）

1) 时钟 RCC

```
RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB, ENABLE);  
RCC_APB1PeriphClockCmd(RCC_APB1Periph_I2C1, ENABLE);
```

2) GPIO 配置

```
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_6 | GPIO_Pin_7;  
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;  
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_OD;    //两个
```

引脚都加 4.7K 上拉电阻

```
GPIO_Init(GPIOB, &GPIO_InitStructure);
```

3) I2C 配置

```
void I2C_Configuration(void)  
{  
    I2C_InitTypeDef I2C_InitStructure;  
  
    I2C_InitStructure.I2C_Mode = I2C_Mode_I2C;  
    I2C_InitStructure.I2C_DutyCycle = I2C_DutyCycle_2;  
    I2C_InitStructure.I2C_OwnAddress1 = 0xc0; // STM32 的自身地址，不与从器件相同即可  
    I2C_InitStructure.I2C_Ack = I2C_Ack_Enable;  
    I2C_InitStructure.I2C_AcknowledgedAddress = I2C_AcknowledgedAddress_7bit;  
    I2C_InitStructure.I2C_ClockSpeed = 100000;  
  
    I2C_Init(I2C1, &I2C_InitStructure);  
    I2C_Cmd(I2C1, ENABLE);  
}
```

至此，STM32 已配置完毕，其实不是那么难。

2、初始化 MPU6050

```
void MPU6050_Initialize()    //初始化过程，其实就是写 5 个寄存器  
{
```

```

    MPU6050_I2C_ByteWrite(0xd0,0x00,MPU6050_RA_PWR_MGMT_1);    //
    reg107, 唤醒, 8M内部时钟源

    MPU6050_I2C_ByteWrite(0xd0,0x07,MPU6050_RA_SMPLRT_DIV);    //
    采用频率 1000
        MPU6050_I2C_ByteWrite(0xd0,0x06,MPU6050_RA_CONFIG);

    MPU6050_I2C_ByteWrite(0xd0,0x01,MPU6050_RA_ACCEL_CONFIG);    //
    加速度量程 2g

    MPU6050_I2C_ByteWrite(0xd0,0x18,MPU6050_RA_GYRO_CONFIG);
    //角速度量程 2000度/s
    }

```

注：0xD0 表示 MPU6050 的地址。我们知道 I2C从器件（在此当然是指 MPU6050）有 8 位的地址，前 7 位由 WHO AM I 确定，第 8 位由 AD0 的电平决定。WHO AM I 默认值是 0x68H（1101000B），AD0 接低电平，所以 MPU6050 的 I2C 地址是 0xD0H（11010000B）。

3、I2C 核心程序（读/写）

1) 写入寄存器

一次写操作分为几个步骤：发送开始信号 -> 起始成功?（可能描述的不太准确） -> 发送 MPU6050 地址、状态（写） -> 写地址成功? -> 发送 MPU6050 内部某个待写寄存器地址 -> 发送成功? -> 发送要写入的内容 -> 发送成功? -> 发送结束信号

总结：先写 MPU6050 地址，再写 寄存器地址，最后写 内容，且每次都要验证（应该和应答信号有关）。这就像寄快递一样，先写市县地址，再写街道地址，最后写门牌号。

再看下面的程序就会更容易理解一些。

```

void MPU6050_I2C_ByteWrite(u8 slaveAddr, u8 pBuffer, u8 writeAddr)
{
    /* Send START condition */
    I2C_GenerateSTART(I2C1, ENABLE);    //发送开始信号
    /* Test on EV5 and clear it */
    while(!I2C_CheckEvent(I2C1, I2C_EVENT_MASTER_MODE_SELECT));

    /* Send MPU6050 address for write */
    I2C_Send7bitAddress(I2C1, slaveAddr, I2C_Direction_Transmitter);
    // 发送 MPU6050 地址、状态（写）
    /* Test on EV6 and clear it */
    while(!I2C_CheckEvent(I2C1,
        I2C_EVENT_MASTER_TRANSMITTER_MODE_SELECTED));

    /* Send the MPU6050's internal address to write to */
    I2C_SendData(I2C1, writeAddr);    //发送 MPU6050内部某个待
    写寄存器地址
    /* Test on EV8 and clear it */
    while(!I2C_CheckEvent(I2C1,
        I2C_EVENT_MASTER_BYTE_TRANSMITTED));

    /* Send the byte to be written */
    I2C_SendData(I2C1, pBuffer);    //发送要写入的内容
    /* Test on EV8 and clear it */
    while(!I2C_CheckEvent(I2C1,
        I2C_EVENT_MASTER_BYTE_TRANSMITTED));

    /* Send STOP condition */
    I2C_GenerateSTOP(I2C1, ENABLE);    //发送结束信号
}

```

```

I2C_GenerateSTART(I2C1, ENABLE); //发送开始信号
}

```

2) 读寄存器

因为 加速度值和角速度值都在寄存器里，所以必须读它才能获得数据。

附上程序

如：MPU6050_I2C_BufferRead(0xd0, receive_data,
MPU6050_RA_WHO_AM_I, 1); //读 WHO_AM_I 寄存器的值 (0x68)
如：

```

void MPU6050_GetRawAccelGyro(s16* AccelGyro)    //读加速度值
和 角速度值
{
    u8 tmpBuffer[14];
    MPU6050_I2C_BufferRead(0xd0, tmpBuffer,
MPU6050_RA_ACCEL_XOUT_H, 14);
    /* Get acceleration */
    for(i=0; i<3; i++)
        AccelGyro[i]=((s16)((u16)tmpBuffer[2*i] << 8) +
tmpBuffer[2*i+1]);
    /* Get Angular rate */
    for(i=4; i<7; i++)                                //在此跳过温度
寄存器，不需要温度值
        AccelGyro[i]=((s16)((u16)tmpBuffer[2*i] << 8) +
tmpBuffer[2*i+1]);
}

```

注：

```

#define MPU6050_RA_ACCEL_XOUT_H    0x3B
#define MPU6050_RA_ACCEL_XOUT_L    0x3C
#define MPU6050_RA_ACCEL_YOUT_H    0x3D
#define MPU6050_RA_ACCEL_YOUT_L    0x3E
#define MPU6050_RA_ACCEL_ZOUT_H    0x3F
#define MPU6050_RA_ACCEL_ZOUT_L    0x40
#define MPU6050_RA_TEMP_OUT_H      0x41
#define MPU6050_RA_TEMP_OUT_L      0x42
#define MPU6050_RA_GYRO_XOUT_H      0x43
#define MPU6050_RA_GYRO_XOUT_L      0x44
#define MPU6050_RA_GYRO_YOUT_H      0x45
#define MPU6050_RA_GYRO_YOUT_L      0x46
#define MPU6050_RA_GYRO_ZOUT_H      0x47
#define MPU6050_RA_GYRO_ZOUT_L      0x48

```

I2C 读核心程序：

```

void MPU6050_I2C_BufferRead(u8 slaveAddr, u8* pBuffer, u8 readAddr,
u16 NumByteToRead)
{
    /* While the bus is busy */
    while(I2C_GetFlagStatus(I2C1, I2C_FLAG_BUSY));

    /* Send START condition */
    I2C_GenerateSTART(I2C1, ENABLE);
    /* Test on EV5 and clear it */
    while(!I2C_CheckEvent(I2C1, I2C_EVENT_MASTER_MODE_SELECT));

    /* Send MPU6050 address for write */
    I2C_Send7bitAddress(I2C1, slaveAddr, I2C_Direction_Transmitter);
    /* Test on EV6 and clear it */
    while(!I2C_CheckEvent(I2C1,
I2C_EVENT_MASTER_TRANSMITTER_MODE_SELECTED));

    /* Clear EV6 by setting again the PF bit */

```

```

/* Clear EV8 by setting again the I2C1 */
I2C_Cmd(I2C1, ENABLE);
/* Send the MPU6050's internal address to write to */
I2C_SendData(I2C1, readAddr);
/* Test on EV8 and clear it */
while(!I2C_CheckEvent(I2C1,
I2C_EVENT_MASTER_BYTE_TRANSMITTED));

/* Send STRAT condition a second time */
I2C_GenerateSTART(I2C1, ENABLE);
/* Test on EV5 and clear it */
while(!I2C_CheckEvent(I2C1, I2C_EVENT_MASTER_MODE_SELECT));

/* Send MPU6050 address for read */
I2C_Send7bitAddress(I2C1, slaveAddr, I2C_Direction_Receiver);
/* Test on EV6 and clear it */
while(!I2C_CheckEvent(I2C1,
I2C_EVENT_MASTER_RECEIVER_MODE_SELECTED));

/* While there is data to be read */
while(NumByteToRead)
{
    if(NumByteToRead == 1)
    {
        /* Disable Acknowledgement */
        I2C_AcknowledgeConfig(I2C1, DISABLE);

        /* Send STOP Condition */
        I2C_GenerateSTOP(I2C1, ENABLE);
    }

    /* Test on EV7 and clear it */
    if(I2C_CheckEvent(I2C1, I2C_EVENT_MASTER_BYTE_RECEIVED))
    {
        /* Read a byte from the MPU6050 */
        *pBuffer = I2C_ReceiveData(I2C1);

        /* Point to the next location where the byte read will be saved */
        pBuffer++;

        /* Decrement the read bytes counter */
        NumByteToRead--;
    }
}

/* Enable Acknowledgement to be ready for another reception */
I2C_AcknowledgeConfig(I2C1, ENABLE);
}

```

最后，把六个值 printf 出来就行。

最后附上程序一份！



[MPU6050 I2C读取数据-精简版.z](#)

4.47 MB, 下载次数: 10871, 下载
积分: 积分 -1 分



收藏 42

标签: [stm32](#) [I2C](#) [MPU6050](#) [stm32](#) [stm32](#)

[举报](#)

相关帖子

- [实习工作总结第二十九天](#)
- [广告贴](#)
- [求助](#)
- [实习工作总结第二十八天](#)
- [stm32 i2c通信的MLX90614测温](#)
- [实习工作总结第二十七天](#)
- [MPU6050移植编译错误](#)

楼主热帖

- [谈一谈 MPU6050 姿态融合](#)
- [时序图绘制软件 —— TimeGen （附件含有软件 + 教...](#)
- [基于STM32F103RCT6的无源蜂鸣器音乐播放（生日快...](#)
- [基于 stm32f103rct6 和 lcd1602 的闹钟程序](#)

发表于 2015-7-19 16:15:47

22[#]

楼主讲的很仔细，谢谢楼主，看了之后明白了很多。。。。。。。

