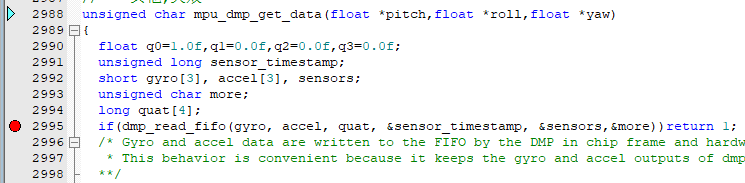
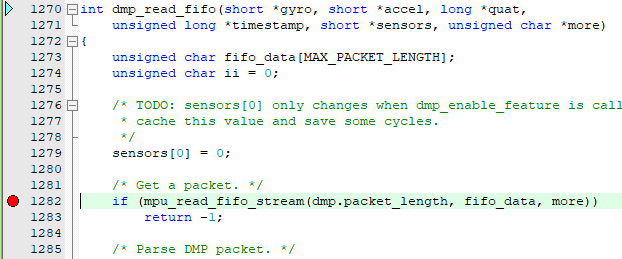
在main函数中使用mpu\_dmp\_get\_data(&pitch, &roll, &yaw)读取欧拉角，发现三个值一直都是0。

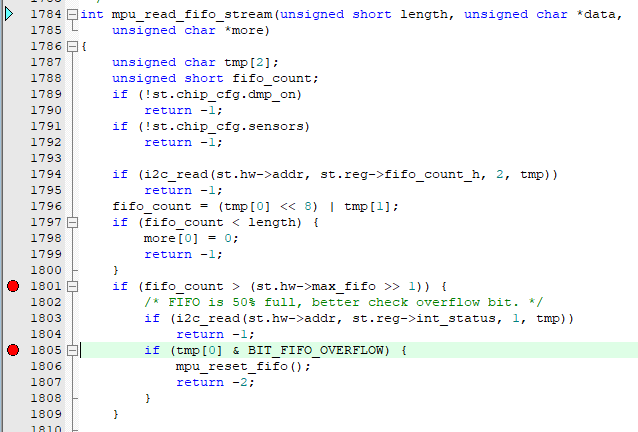
通过debug模式跟踪，发现在其dmp\_read\_fifo(gyro, accel, quat, &sensor\_timestamp, &sensors,&more)这里就返回了，



继续跟踪到mpu\_read\_fifo\_stream(dmp.packet\_length, fifo\_data, more)就返回了，

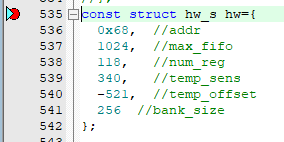


再跟踪到if(fifo\_count > (st.hw->max\_fifo >> 1))这里，返回-2，



这一段是检测如果fifo已有的数据超过设定fifo容量的一半时，会获取int\_status寄存器状态，并判断BIT\_FIFO\_OVERFLOW这个位是否被置位，被置位就说明FIFO已满，它会清空FIFO并放弃本次读取。

调试到这里时，fifo\_count的值为0x400，而st.hw->max\_fifo的值设置也是0x400(1024)，FIFO确实已经满了。



既然已经知道FIFO是满了才导致读取FIFO失败，那就找出FIFO会满的原因。

原来是main循环里一开始写了个HAL\_Delay(500)，MPU初始化后延时太久没读取FIFO，导致FIFO满载，将这条延时屏蔽掉，就能正常地读取欧拉角。

也就是说解决这个问题需要提高读取FIFO的频率，及时从FIFO读取数据，保持FIFO不满载状态。

当然网上还有些其他解决方案，有的说降低MPU的采样率（默认100HZ），有的说将if (tmp[0] & BIT\_FIFO\_OVERFLOW)这段屏蔽就好了，就是不对FIFO满载作处理。

