Hi,
I try to use IRQ to read data from MPU 6050, but event I2C_EVENT_MASTER_BYTE_RECEIVED is never trigered.
The interrupt code looks like following

```
void I2C1_EV_IRQHandler(void)
{
    volatile static uint8_t first_I2C_EVENT_MASTER_BYTE_TRANSMITTED = 1;

    uint32_t event = ( (uint32_t)(I2C1->SR1) | (uint32_t)(I2C1->SR2)<<16 ) & 0x00FFFFFF;

    if (gv_evlog_pointer < 63)
    {
        gv_evlog[gv_evlog_pointer++] = event;
    }

    switch (event)
    {

        case I2C_EVENT_MASTER_MODE_SELECT:  // EV5
            first_I2C_EVENT_MASTER_BYTE_TRANSMITTED = 1;
            if ( gv_i2c1_phase == I2C_PHASE_REG_NUMBER_SENT )
            {
                // sending addres and set reciver mode
                I2C_Send7bitAddress(MPU6050_I2C_DEV, MPU6050_I2C_ADDRESS, I2C_Direction_Receiver);
            }else
            {
                // sending addres and set transmiter mode
                I2C_Send7bitAddress(MPU6050_I2C_DEV, MPU6050_I2C_ADDRESS,
I2C_Direction_Transmitter);
            }
            break;

        case I2C_EVENT_MASTER_TRANSMITTER_MODE_SELECTED: // EV6
              /* Clear EV6 by setting again the PE bit */
             I2C_Cmd(MPU6050_I2C_DEV, ENABLE);

            // sending register number
            I2C_SendData(MPU6050_I2C_DEV, gv_i2c1_reg);
            gv_i2c1_phase = I2C_PHASE_REG_NUMBER_SENT;
            break;

        case I2C_EVENT_MASTER_BYTE_TRANSMITTED: // EV8
            if (first_I2C_EVENT_MASTER_BYTE_TRANSMITTED == 1)
                // restart for slave transmistion
                I2C_GenerateSTART(MPU6050_I2C_DEV, ENABLE);
                first_I2C_EVENT_MASTER_BYTE_TRANSMITTED = 0;
            break;

        case I2C_EVENT_MASTER_RECEIVER_MODE_SELECTED: // EV 6
            // disable acknowledge
            I2C_AcknowledgeConfig(MPU6050_I2C_DEV, DISABLE);
             /* Send STOP Condition */
            I2C_GenerateSTOP(MPU6050_I2C_DEV, ENABLE);

            break;

        case I2C_EVENT_MASTER_BYTE_RECEIVED: // EV 7
            gv_i2c1_reg_val = I2C_ReceiveData(MPU6050_I2C_DEV);
            // generate stop
            I2C_GenerateSTOP(MPU6050_I2C_DEV, ENABLE);
            // turn on acknowlage
            I2C_AcknowledgeConfig(MPU6050_I2C_DEV, ENABLE);
            gv_i2c1_phase = I2C_PHASE_BYTE_RECIVED;
            I2C_ITConfig(MPU6050_I2C_DEV, I2C_IT_EVT, DISABLE);
            break;
```

```
        default:
            // test
            qconsole_led_blink(QCONSOLE_LED_B1, QCONSOLE_LED_SPEED_FAST, 10);
            break;
    }
}
```

The read register function is

```
uint8_t qmpu6050_read_register_irq(uint8_t reg, uint8_t * val)
{

    uint32_t event;
    printf("qmpu6050_read_register_irq\n\r");

    // wait for i2c device
    //while (I2C_GetFlagStatus(MPU6050_I2C_DEV, I2C_FLAG_BUSY));
    while ( ( event = qmpu6050_get_event_flags() ) & I2C_FLAG_BUSY )
    {
        printf("I2C_FLAG_BUSY %x\n\r", event);
    }

    // store data to global variables
    gv_i2c1_reg = reg;
    gv_i2c1_phase = I2C_PHASE_DEVICE_READY;
    gv_evlog_pointer = 0;
    memset(gv_evlog, 0, 64);

    I2C_ITConfig(MPU6050_I2C_DEV, I2C_IT_EVT, ENABLE);
    //transmision start

    I2C_GenerateSTART(MPU6050_I2C_DEV, ENABLE);



    delay_ms(500);

    for (int i=0; i < 64; i++)
    {
        printf("Event no %d = %x\n\r", i, gv_evlog[i]);
    }

    while (gv_i2c1_phase != I2C_PHASE_BYTE_RECIVED)
    {
    //  printf("Wait for BYTE_RECIVED_PHASE %x Event=%x\n\r", gv_i2c1_phase, gv_event);

    }
    * val = gv_i2c1_reg_val;

    return QMPU6050_RET_OK;
}
```

The gv_evlog variable contains following entries
Event no 0 = 30001 I2C_EVENT_MASTER_MODE_SELECT
Event no 1 = 70082 I2C_EVENT_MASTER_TRANSMITTER_MODE_SELECTED
Event no 2 = 70084 I2C_EVENT_MASTER_BYTE_TRANSMITTED
Event no 3 = 70084 I2C_EVENT_MASTER_BYTE_TRANSMITTED
Event no 4 = 70084 I2C_EVENT_MASTER_BYTE_TRANSMITTED
Event no 5 = 70084 I2C_EVENT_MASTER_BYTE_TRANSMITTED
Event no 6 = 70084 I2C_EVENT_MASTER_BYTE_TRANSMITTED
Event no 7 = 70084 I2C_EVENT_MASTER_BYTE_TRANSMITTED
Event no 8 = 70084 I2C_EVENT_MASTER_BYTE_TRANSMITTED
Event no 9 = 70084 I2C_EVENT_MASTER_BYTE_TRANSMITTED

Event no 10 = 70084 I2C_EVENT_MASTER_BYTE_TRANSMITTED
Event no 11 = 70084 I2C_EVENT_MASTER_BYTE_TRANSMITTED
Event no 12 = 70084 I2C_EVENT_MASTER_BYTE_TRANSMITTED
Event no 13 = 30001 I2C_EVENT_MASTER_MODE_SELECT
Event no 14 = 30002 I2C_EVENT_MASTER_RECEIVER_MODE_SELECTED

so, the I2C_EVENT_MASTER_BYTE_RECEIVED is never trigered :(

Similar code using pooling working ok (except limitation described in errata).

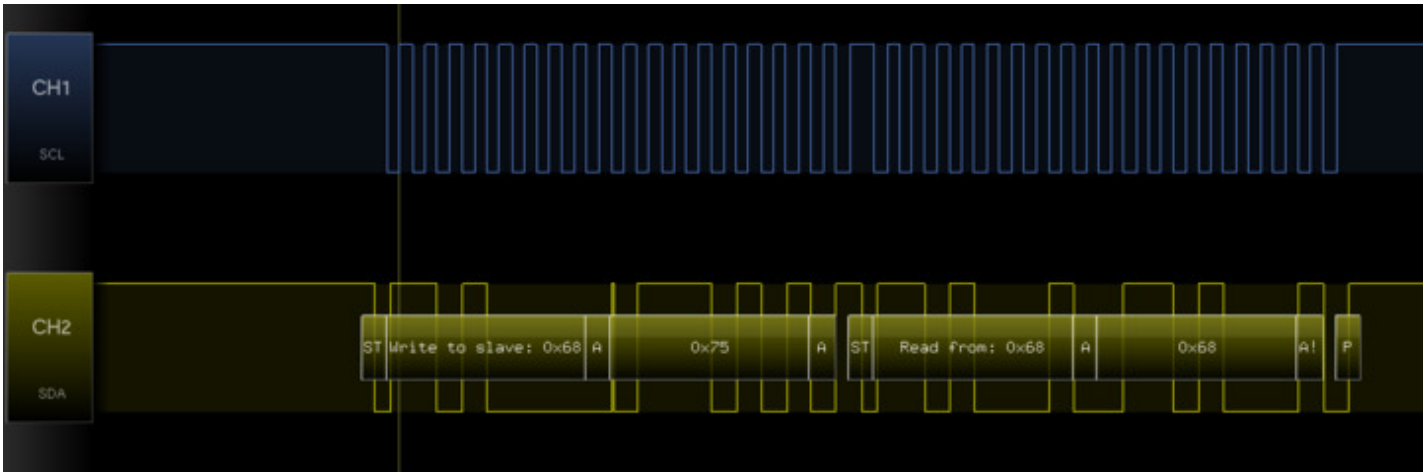
```
uint8_t qmpu6050_read_register(uint8_t reg, uint8_t * val)
{

    uint32_t event;

    //while (I2C_GetFlagStatus(MPU6050_I2C_DEV, I2C_FLAG_BUSY));
    while ( ( event = qmpu6050_get_event_flags() ) & I2C_FLAG_BUSY )
    {
        //printf("I2C_FLAG_BUSY");
    }
    //transmision start

    I2C_GenerateSTART(MPU6050_I2C_DEV, ENABLE);

    //wait for ev5

    //while( !I2C_CheckEvent(MPU6050_I2C_DEV, I2C_EVENT_MASTER_MODE_SELECT))
    while ( ( ( event = qmpu6050_get_event_flags() ) & I2C_EVENT_MASTER_MODE_SELECT ) !=
I2C_EVENT_MASTER_MODE_SELECT )
    {
        //printf("I2C_EVENT_MASTER_MODE_SELECT %x\n\r", event);
    }

    // send mpu6050 address

    I2C_Send7bitAddress(MPU6050_I2C_DEV, MPU6050_I2C_ADDRESS, I2C_Direction_Transmitter);

    // wait for ev6

    //while( !I2C_CheckEvent(MPU6050_I2C_DEV, I2C_EVENT_MASTER_TRANSMITTER_MODE_SELECTED))
    while ( ( ( event = qmpu6050_get_event_flags() ) & I2C_EVENT_MASTER_TRANSMITTER_MODE_SELECTED )
!= I2C_EVENT_MASTER_TRANSMITTER_MODE_SELECTED )
    {
        //printf("I2C_EVENT_MASTER_TRANSMITTER_MODE_SELECTED %x\n\r", event);
    }

    /* Clear EV6 by setting again the PE bit */
    I2C_Cmd(MPU6050_I2C_DEV, ENABLE);

    // send register no
    I2C_SendData(MPU6050_I2C_DEV, reg);

    // wait for ev8

    //while( !I2C_CheckEvent(MPU6050_I2C_DEV, I2C_EVENT_MASTER_BYTE_TRANSMITTED))
    while ( ( ( event = qmpu6050_get_event_flags() ) & I2C_EVENT_MASTER_BYTE_TRANSMITTED ) !=
I2C_EVENT_MASTER_BYTE_TRANSMITTED )
        {
            //printf("I2C_EVENT_MASTER_BYTE_TRANSMITTED %x\n\r", event);
        }

    // send restart for starting slave transmision
    I2C_GenerateSTART(MPU6050_I2C_DEV, ENABLE);

    //wait for ev5
```

```
    //while( !I2C_CheckEvent(MPU6050_I2C_DEV, I2C_EVENT_MASTER_MODE_SELECT))
    while ( ( ( event = qmpu6050_get_event_flags() ) & I2C_EVENT_MASTER_MODE_SELECT) !=
I2C_EVENT_MASTER_MODE_SELECT)
        {
            //printf(" I2C_EVENT_MASTER_MODE_SELECT %x\n\r", event);
        }

    // send mpu6050 address

    I2C_Send7bitAddress(MPU6050_I2C_DEV, MPU6050_I2C_ADDRESS, I2C_Direction_Receiver);


    // wait for ev6

    //while( !I2C_CheckEvent(MPU6050_I2C_DEV, I2C_EVENT_MASTER_RECEIVER_MODE_SELECTED))
    while ( ( ( event = qmpu6050_get_event_flags() ) & I2C_EVENT_MASTER_RECEIVER_MODE_SELECTED ) !=
I2C_EVENT_MASTER_RECEIVER_MODE_SELECTED )
        {
            //printf("I2C_EVENT_MASTER_RECEIVER_MODE_SELECTED %x\n\r", event);
        }

    // disable acknowledge

    I2C_AcknowledgeConfig(MPU6050_I2C_DEV, DISABLE);

    /* Send STOP Condition */
    I2C_GenerateSTOP(MPU6050_I2C_DEV, ENABLE);

    //wait for ev7

    //while( !I2C_CheckEvent(MPU6050_I2C_DEV, I2C_EVENT_MASTER_BYTE_RECEIVED))
    while ( ( ( event = qmpu6050_get_event_flags() ) & I2C_EVENT_MASTER_BYTE_RECEIVED ) !=
I2C_EVENT_MASTER_BYTE_RECEIVED )
        {
            //printf("I2C_EVENT_MASTER_BYTE_RECEIVED %x\n\r", event);
        }

    // get byte

    * val = I2C_ReceiveData(MPU6050_I2C_DEV);

    // generate stop

    I2C_GenerateSTOP(MPU6050_I2C_DEV, ENABLE);

    // turn on acknowlage

    I2C_AcknowledgeConfig(MPU6050_I2C_DEV, ENABLE);


    return QMPU6050_RET_OK;
}
```

What am i doing wrong? :)
Thank you in advance

Krzysiek

PS
This is i2c transfer, and it looks quite good


CH1
SCL
CH2
SDA
ST
Write to slave: 0x68
A
0x75
A
ST
Read from: 0x68
A
0x68
A!
P