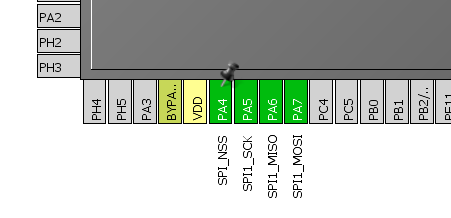
# ADIS16470测评报告

前一段时间拿到了大疆给的高性能陀螺仪ADIS16470，迫不及待的就开始了陀螺仪的测试，但是在测评的过程中遇到了各种小问题导致迟迟不能得到较好的结果，在经历了不断的探索后最终拿到了想要的数据，在此分享该过程中需要注意的问题和最终的成果和本例程。

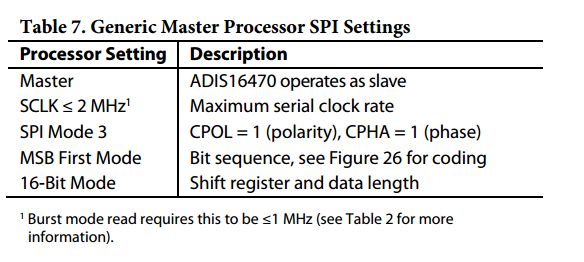
## 基本程序设置与电路搭建

通过数据手册可以知道，ADIS16470是通过SPI的方式来与主设备的通信的，组成SPI通信需要CLK时钟线，MOSI输出线，MISO输入线与NSS片选线。我手头上就只有一个STM32F439IGT6核心板，所以我的工程选用了该板来建立工程。

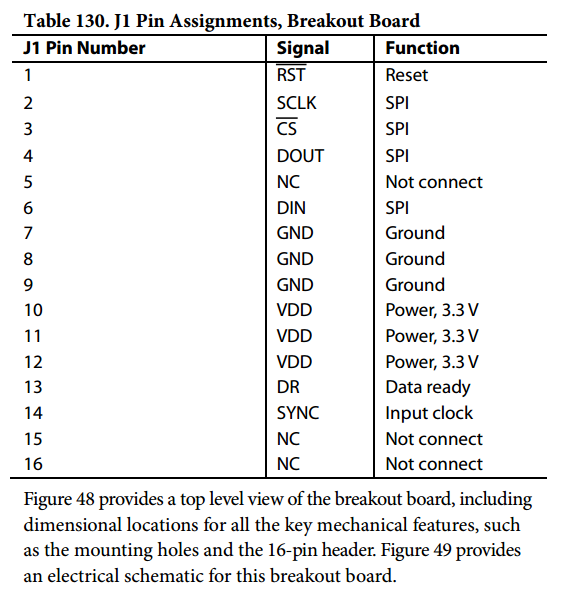
在STM32CubeMX中建立STM32F439IGT6的工程后，建立工程后首先设定好时钟频率和调试方式，然后启用一个SPI和一个普通GPIO作为NSS片选线，如图:   


然后是配置SPI和GPIO方式，作为SPI信号线，GPIO建议设置为开漏输出模式

SPI的配置方式要根据数据手册来决定，查阅数据手册可以发现，CPOL应为1，CPHA应为1，在Cube中即设置Polarity=HIGH， Phase=2Edge（这个问题在刚开始弄的时候一不小心弄错了 折腾了很久才弄出来）。同时因为ADIS16470传输数据是以16bits为一个单位的，可以在Cube中设置SPI的Data Size=16。然后还要调整SPI的Prescaler以降低波特率以保证不超过2Mb/s(使用BurstRead时不能超过1Mb/s)

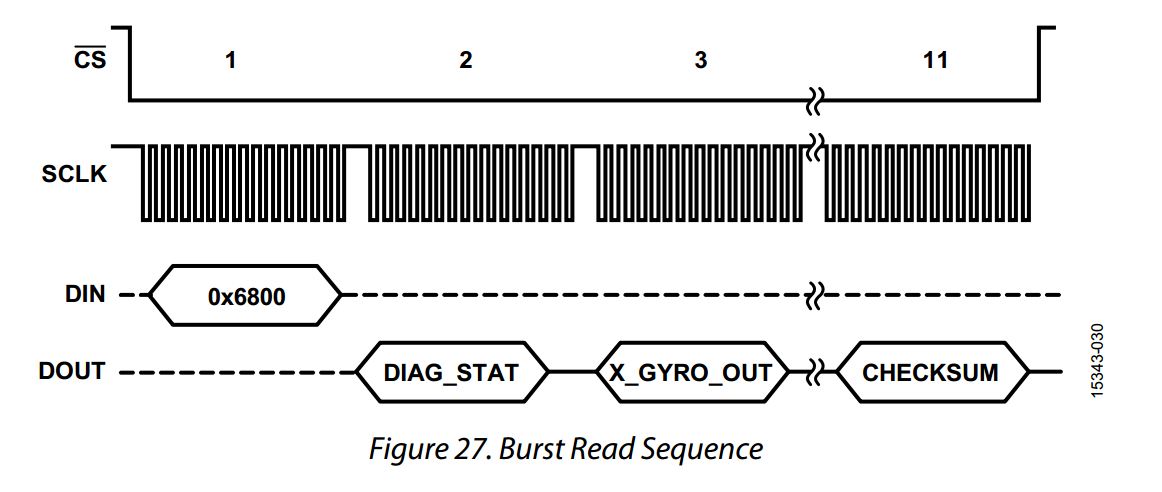


当以上配置都完成之后可以生成工程，然后根据陀螺仪的数据手册中的引脚定义，进行核心板和陀螺仪的接线。这里有一点其实我听迷惑的，就是这里J1的序号并不是从上到下排序的，而是按照行来进行编号的，这一点大家要注意，别一不小心误操作就把这贵重的陀螺仪给烧了!

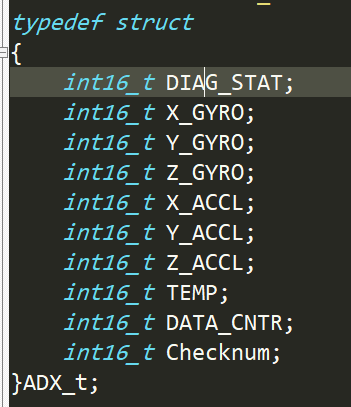


## 以BurstRead方式读取原始数据

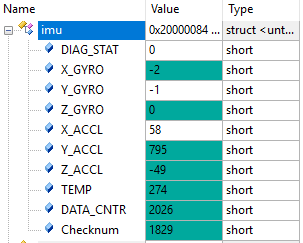
该陀螺仪的读取数据的方式有很多种，而BurstRead就是其中一种方式，它的特点是读取简单，连续读取，一步到位获得三轴加速度和三轴角速度值。在对精度要求不是很高要求快速上手的情况下，BurstRead方式是一个不错的选择。



查阅技术手册可以得到BurstRead方式的操作时序图，在启动的时候将CS线拉低，随后发送一个字0x6800，然后连续读取SPI数据，就能得到按照一定顺序排列的传感器数据了。

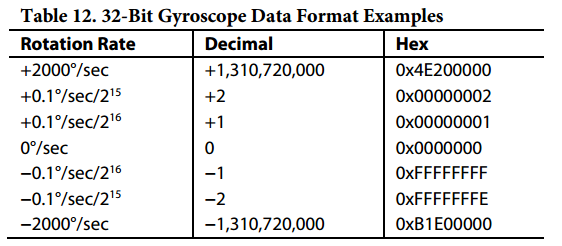
根据数据手册上数据的排列，可以定义一个结构体用于存放数据: 

然后就可以在程序中做一个循环读取，读取前先发BurstRead前置指令，然后将剩下读到的数据放到结构体内即可。为了保证数据可靠性，每一帧数据都存在一个校验字节可用于保证帧完整性和可靠性。在每次接收完完整的一帧之后，可以进行数据校验，仅当校验成功时保留数据。以下为读取出来的传感器值(原始数据值):

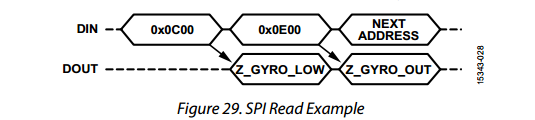


## 通过单独读取寄存器来获得传感器数据

在大部分时候，BurstRead可以满足基本上的需求，可是ADIS16470拥有更精准的32位精度的数据，在我们有更高要求的时候，通过读取其寄存器可以得到非常高的精度的传感器数据。

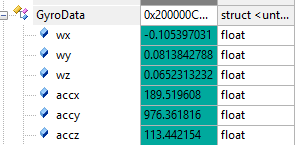
下图则为其角速度的32位数据精度，为655360LSB每度每秒，可以说是十分吓人了:

所以，当我们有高精度需求的时候，就需要单独的去读取它的寄存器了。读取寄存器的时序如下:

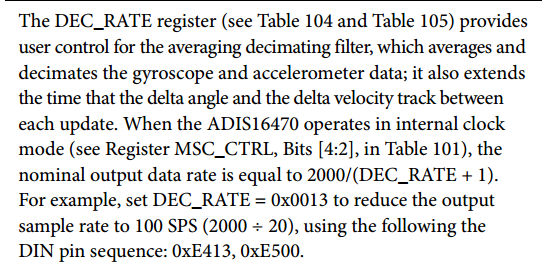


可以看到，每次发送需要读取的寄存器的时候，读取结果会在SPI传输的下一个时序中返回。为了提高读取效率，可以将需要读取的寄存器地址连续发送，然后在接收时加上偏移即可。

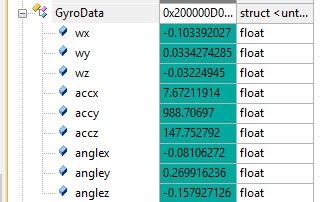
在读取完毕之后，将两个16位数据合成一个32位的数据(根据手册，按照小端模式，低位低地址高位高地址合并)，这样就得到了32位的原始数据。仅仅得到原始数据还不够，根据如上图的表格得到原始数据到标准单位的关系进行换算，最终得到三轴角速度以及三轴加速度值。



在取到三轴角速度和三轴加速度后，我们还有一个经常使用的值，就是三轴的姿态角。很多时候对于普通陀螺仪我们都是直接取速度的积分作为角度值，这样存在一定的偏差。为了解决积分的准确度，ADIS16470还提供了角度差数据。这个数据提供一定时间区间内的角度差，默认值为1/2000秒的角度差。在做机器人控制系统的时候，我们一般不需要用到这么高的数据频率，因此我们需要提高积分的时间，即降低数据频率。ADIS16470中提供了一个寄存器用来控制积分时间，如果这个寄存器为x的话，数据频率就为2000/(x+1)Hz。该寄存器的默认值为0，所以默认频率为2000Hz。在控制系统的用途中，我们把数据频率控制到500Hz就可以获得较佳的控制效果了。因此我们需要写入该寄存器为0x03把积分时间提升到1/500s。

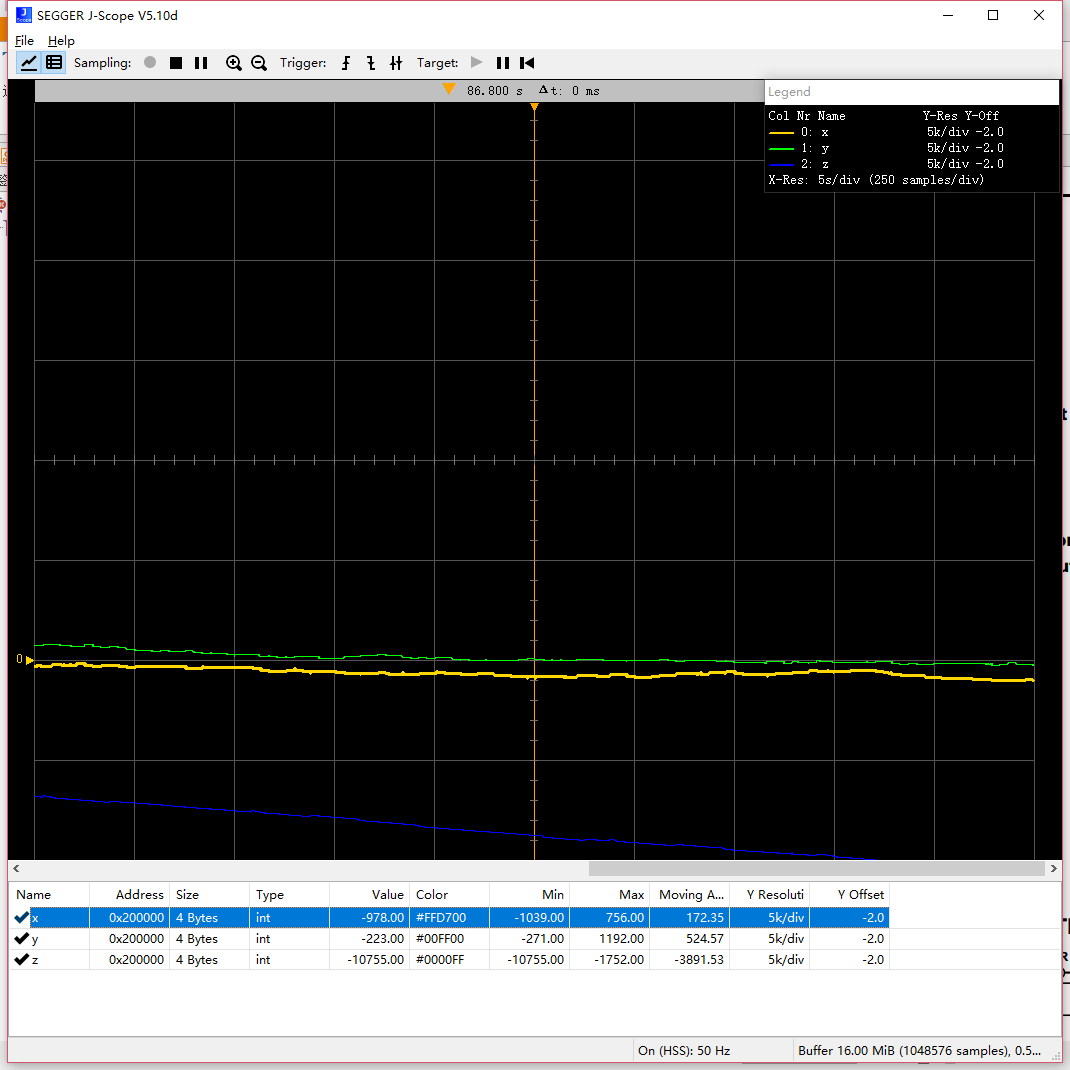


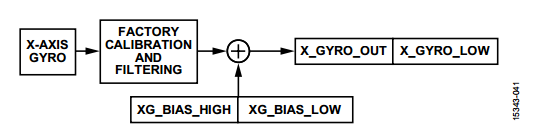
当我们的控制周期为500Hz时，每次读取角度差值并进行积分，此时积分的结果就是较为精确的角度值，不会因为控制频率的降低而损失精度。 最终的所有数据如下:



## 简单的数据处理

从刚刚我们已经得到了所有的数据，不过数据还是不大稳定，这时候还需要做一些处理。

陀螺仪由于其硬件限制，会存在零点漂移，也就是说当真实的某轴角速度为0的时候，其输出的值并不是0而是一个接近0的数，也就是说它的零点发生了偏移。ADIS16470内部提供了零点校准的寄存器值，其工作的原理如下：



也就是说，确定好了合适的零点偏移量后，每次陀螺仪的数据都会经过偏移量的修正从而得到较为准确的数据。一般常用确定零点漂移量的方法是在程序启动的一段时间内保持陀螺仪不发生运动，采集一段时间的数据，然后用得到的角度除以自检的时间得到偏移角速度，然后将其取负号作为零点漂移修正值。在进行零点修正后，陀螺仪就能读出更稳定的数据了。

当然影响陀螺仪的准确度的因素远不止这一种，还有很多诸如随机漂移，高频振动等各种影响陀螺仪准确度的因素，不过因为ADIS16470的各个方面的性能都很出色，把像随机漂移和高频振动带来的偏移这种较难处理的误差降到很低，所以简单处理就能得到较好的数据。除了零点修正外，ADIS16470还内置了巴特沃斯滤波器，可以根据需求通过配置寄存器可以开启滤波器得到更准确的数据。如果还需要更精确的数据还可以考虑数据融合算法，在此不再赘述。