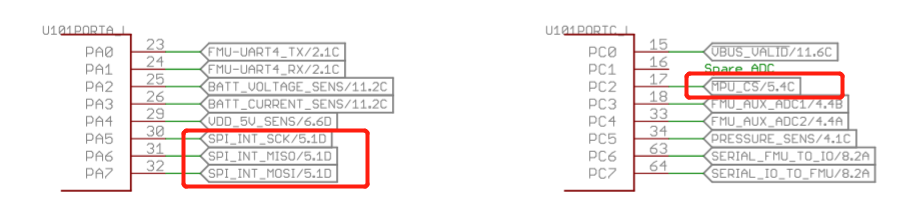
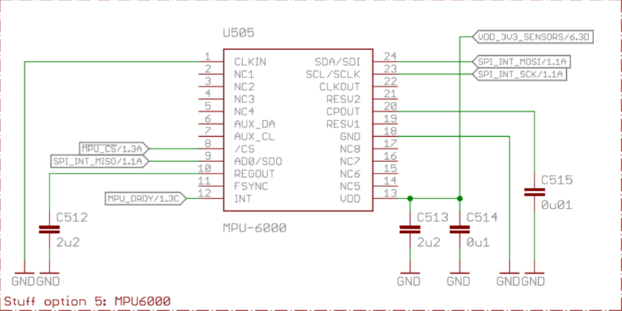
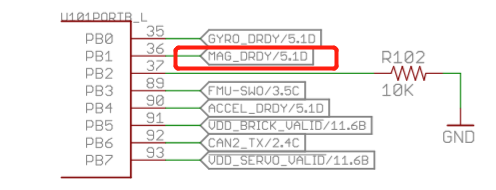
SPI与传感器：读取IMU数据实验

1. **硬件说明**
2. **SPI相关的硬件说明**



（1） （2）



（3） （4）

图1. SPI相关硬件说明

如图1所示，MPU6000通过SPI与STM32F427芯片通信。引脚连接如表1.所示。

表1. MPU6000与STM32F427的引脚连接

|  |  |  |
| --- | --- | --- |
| STM32F427 | MPU6000 | 用途 |
| PA5 | SCLK | SPI的时钟 |
| PA6 | SDO | SPI的MISO |
| PA7 | SDI | SPI的MOSI |
| PC2 | CS | 片选信号输入 |
| PB1 | INT | 数字中断信号输出 |

1. **飞控板的IMU硬件说明**

飞控板上的IMU芯片为MPU6000，可以通过IIC和SPI两种通信协议读取，能够输出3轴加速度值，3轴陀螺仪数据，以及温度数据，其硬件说明参考[技术手册](../datasheet/MPU6000.pdf)。

本实验中我们需要写MPU6000的驱动函数，需要参考芯片的[寄存器手册](../datasheet/MPU-6000-Register-Map1.pdf)。

1. **工程配置**
2. **SPI配置**

SPI配置如图2. 所示。根据硬件原理图，我们需要配置STM32F4的PA5，PA6，PA7三个与SPI相关的引脚，这三个引脚对应的是STM32芯片的SPI1。

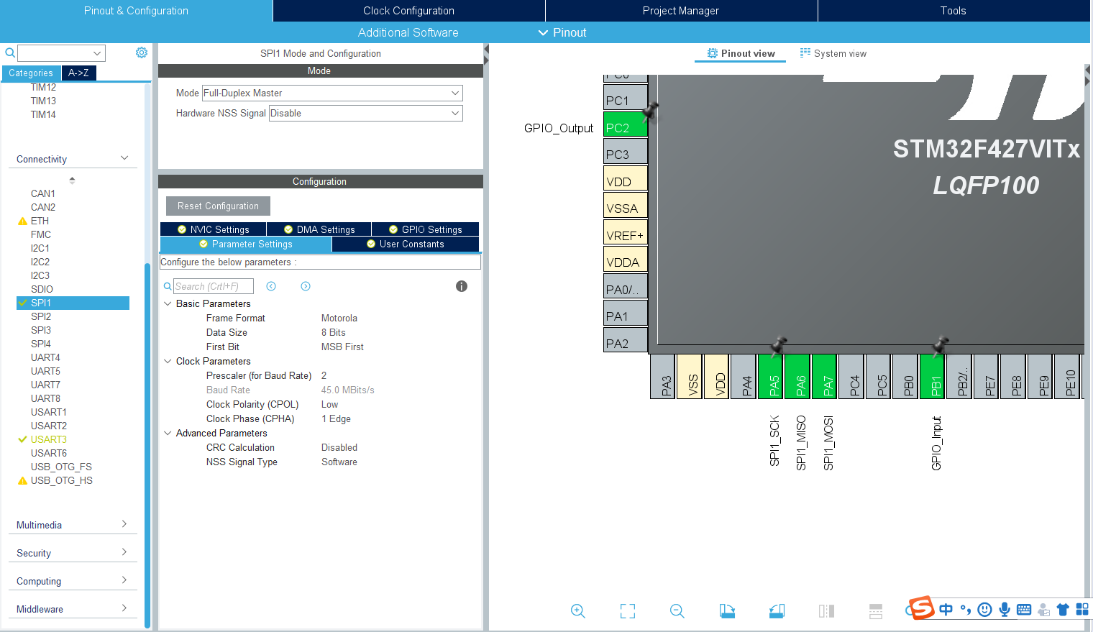
****

图2. SPI配置

1. **添加自己的库函数**

在之前的例程中，例如串口，ADC，定时器，我们都是在库函数中直接调用的HAL库中的函数，完成一些功能。但是HAL库中的函数很可能并不能够完全实现我们所需要的功能，所以在大多数时候，我们需要基于HAL库来完成自己的库函数的编写。

举个例子，HAL库中有SPI的接收和发送的函数，如下表所示：

|  |
| --- |
| HAL\_StatusTypeDef **HAL\_SPI\_TransmitReceive**(SPI\_HandleTypeDef \*hspi, uint8\_t \*pTxData, uint8\_t \*pRxData, uint16\_t Size,uint32\_t Timeout); |

这个函数中包含5个形参，但是在我们的实际的应用中，如果每次都要输入5个形参调用HAL库，是非常繁琐的。而且大多数时候，也并不需要每次都设置5个形参，比如在通过SPI给寄存器写入值得时候，按照我们惯常的思维，只需要设计一个函数，形参包括寄存器的地址，要写入的数据值就可以了。我们可以定义一个函数，如下表所示：

|  |
| --- |
| **void** **Spi\_GyroSingleWrite**(uint8\_t reg, uint8\_t value); |

这样的函数，方便调用，而且符合人们的惯常编程思维。因此，将HAL库封装为我们需要的库函数的过程，就被称为添加自己的库函数。

一般来讲，每一个单独的功能，需要包含一个”.c”文件，和一个“.h”文件，如图3所示为本实验例程的文件架构：

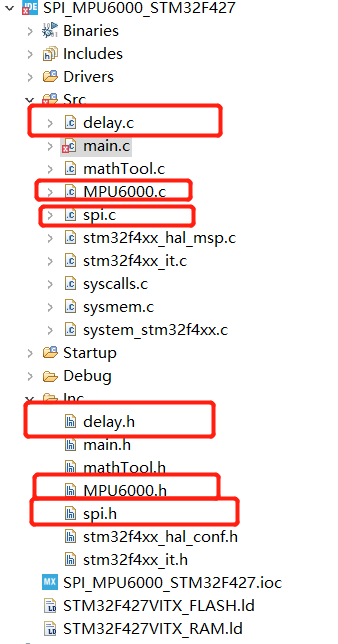


图3. 本实验的文件架构

SPI，MPU6000的驱动程序，延时程序，可以划分为三个不同的模块，每个模块包含不同一对单独的.c和.h文件，图中红框所示。向工程中，添加自己的文件，可以通过图4. 的方式完成。

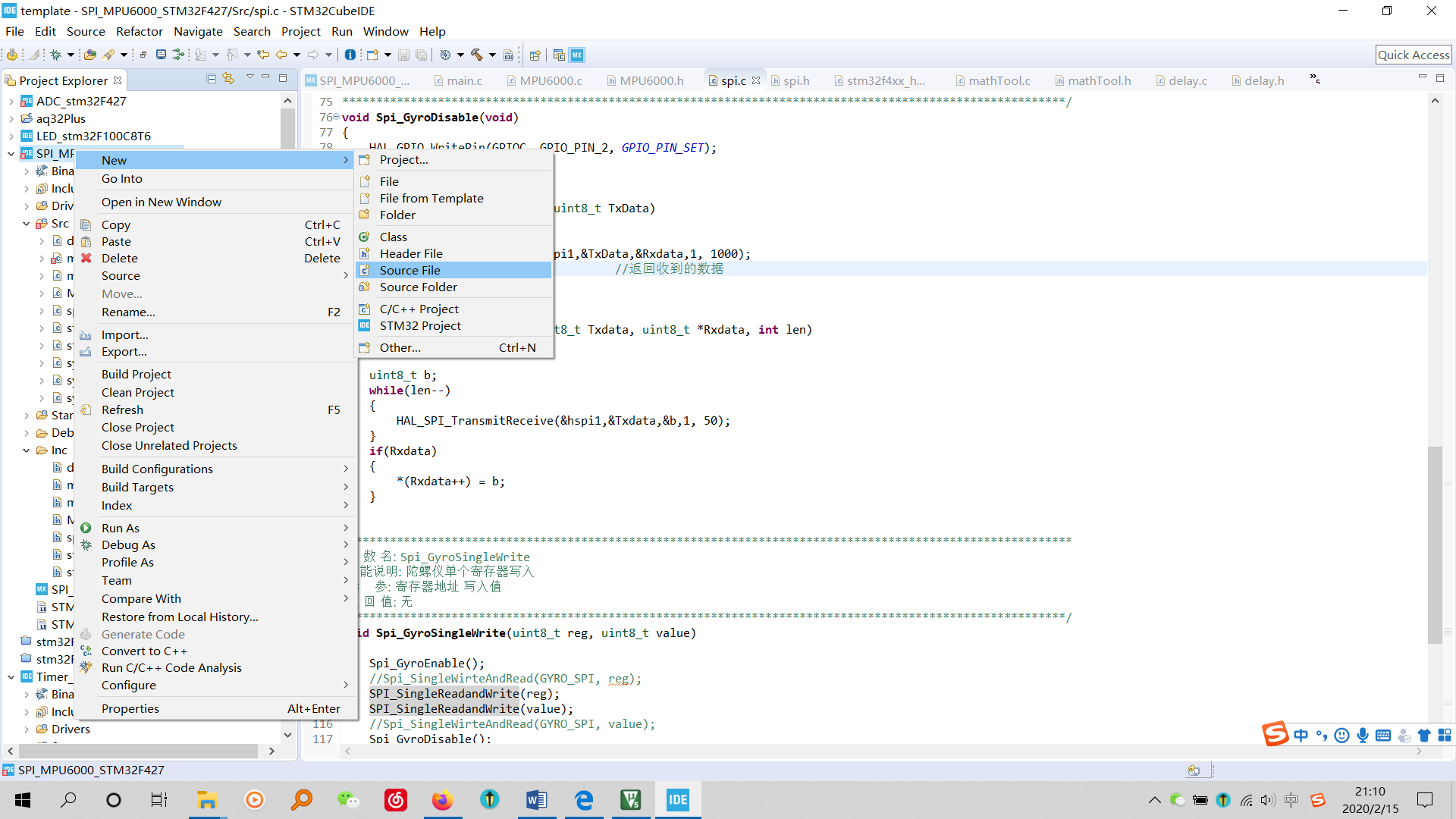


图4. 添加问价的方式

具体的驱动程序，SPI程序，延时程序，在样例代码中去详细了解，在此不做过多的解释。

1. **主程序的编写**

主函数中，我们实现的功能是，每间隔一秒钟，单片机向上位机发送MPU6000传感器采集到的温度数据。

|  |
| --- |
| **#include** "main.h"  **#include** "spi.h"  **#include** "MPU6000.h"  UART\_HandleTypeDef huart3;  **void** **SystemClock\_Config**(**void**);  **static** **void** **MX\_GPIO\_Init**(**void**);  **static** **void** **MX\_USART3\_UART\_Init**(**void**);  **int** **main**(**void**)  {    HAL\_Init();  SystemClock\_Config();  MX\_GPIO\_Init();  MX\_SPI1\_Init();  MX\_USART3\_UART\_Init();  MX\_TIM2\_Init();  MPU6000\_Init();  /\* USER CODE BEGIN WHILE \*/  **while** (1)  {  //每一秒，输出一次温度的数据  /\* USER CODE END WHILE \*/  /\* USER CODE BEGIN 3 \*/  **float** temp;  HAL\_Delay(1000);  MPU6000\_ReadTemp(&temp);  uint8\_t pData[4]="";  **sprintf**(pData,"%.1f",temp);  HAL\_UART\_Transmit(&huart3,(uint8\_t\*)pData,**sizeof**(pData),100);//不断通过串口3发送，pData中的数据，sizeof计算pData中的字节数  }  /\* USER CODE END 3 \*/  } |

1. **硬件连接**

此实验需要使用串口进行通信，连接方式与实验7中的连接方式相同，在程序执行之后，可以显示出芯片的温度值，效果图与实验9中相似，数值显示为ASCII码显示。

1. **延伸**
2. 修改while（1）中的代码，实现每间隔1秒钟，发送三轴加速度，三轴陀螺仪数据的功能。
3. 理解MPU6000传感器驱动程序的编程思路，进而推广到所有传感器驱动程序的编程思路。
4. 传感器采集到的数据就可以直接应用吗？还需不需要进行一些其他的处理？如果需要，通常要对传感器采集到的原始数据做哪些处理？