**作业七（I2C、SPI）**

班级 ，姓名 ，学号 ，成绩

要求：1、A4纸正反打印，手工、独立、认真完成，严禁抄袭。

2、按学号顺序收交

1、回答下列I2C相关问题：

1）I2C的英文全称是什么，是哪个公司提出的串行通信协议

2）I2C是几线式通信协议，每根信号线的作用是什么？

3）连接在I2C上的器件有哪几种工作模式？

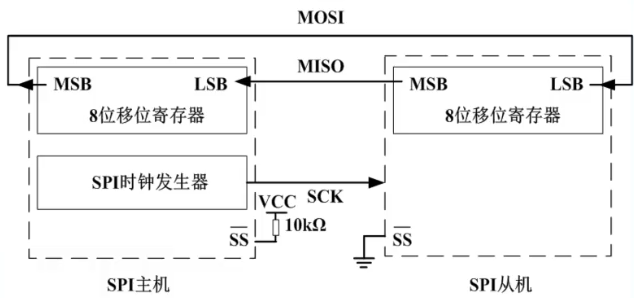
4）I2C实现的是单工通信，还是半双工、全双工通信？同步还是异步？

5）I2C是如何对器件进行寻址的？

2、简述I2C协议的具体内容，包括空闲状态、开始信号、停止信号、应答信号、数据有效性、数据传输等方面的要求。

3、SPI的英文全称是什么，是哪个公司提出的串行通信协议？SPI是几线式通信协议，每根信号线的作用是什么？

4、下图是SPI总线内部结构简图，根据此图简述SPI总线的原理。



5、下面代码将STM32F4的SPI2初始化为2线全双工、16位数据格式、LSB在先、时钟空闲电平位低电平，第一时钟边沿采集数据的主模式，NSS采用软件方式管理，时钟分频系数为64，采用CRC校验。请将代码补充完整。

void SPI2\_Init(void) {

GPIO\_InitTypeDef GPIO\_InitStructure; SPI\_InitTypeDef SPI\_InitStructure;

RCC\_AHB1PeriphClockCmd(RCC\_AHB1Periph\_GPIOB, ENABLE);//GPIO时钟

RCC\_APB1PeriphClockCmd(RCC\_APB1Periph\_SPI2, ENABLE); //SPI2时钟，APB1上，PB10/PB13-SCK、PB14/PC2-MISO、PB15/PC3-MISO

GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_10|GPIO\_Pin\_14|GPIO\_Pin\_15;

GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_AF;

GPIO\_InitStructure.GPIO\_OType = GPIO\_OType\_PP;

GPIO\_InitStructure.GPIO\_Speed = GPIO\_Speed\_100MHz;

GPIO\_InitStructure.GPIO\_PuPd = GPIO\_PuPd\_UP;

GPIO\_Init( GPIOB , &GPIO\_InitStructure);

GPIO\_PinAFConfig( GPIOB , GPIO\_PinSource10 ,GPIO\_AF\_SPI2);

GPIO\_PinAFConfig( GPIOB , GPIO\_PinSource14 ,GPIO\_AF\_SPI2);

GPIO\_PinAFConfig( GPIOB , GPIO\_PinSource15 ,GPIO\_AF\_SPI2);

RCC\_APB1PeriphResetCmd(RCC\_APB2Periph\_SPI2,ENABLE);

RCC\_APB1PeriphResetCmd(RCC\_APB2Periph\_SPI2,DISABLE);

SPI\_InitStructure.SPI\_Direction = SPI\_Direction\_2Lines\_FullDuplex; //2线全双工

SPI\_InitStructure.SPI\_Mode = SPI\_Mode\_Master; //主模式

SPI\_InitStructure.SPI\_DataSize = SPI\_DataSize\_16b; //16位

SPI\_InitStructure.SPI\_CPOL = SPI\_CPOL\_Low ; //空闲电平为低

SPI\_InitStructure.SPI\_CPHA = SPI\_CPHA\_1Edge; //第1边沿采集数据

SPI\_InitStructure.SPI\_NSS = SPI\_NSS\_Soft; //NSS软件管理

SPI\_InitStructure.SPI\_BaudRatePrescaler = SPI\_BaudRatePrescaler\_64; //64分频

SPI\_InitStructure.SPI\_FirstBit = SPI\_FirstBit\_LSB; //LSB在先

SPI\_InitStructure.SPI\_CRCPolynomial = 7; //CRC校验>1

SPI\_Init(SPI2, &SPI\_InitStructure);

SPI\_Cmd(SPI2, ENABLE);

SPI2\_ReadWriteByte(0xff);

}