

软件设计第5讲

哈尔滨工业大学（深圳）
实验与创新实践教育中心

主要内容

- ➡ ◆ 串行通信接口SPI模式
- ◆ 展馆灯光综合调试
- ◆ 项目制作准备

串行通信接口SPI模式 (Serial Peripheral Interface)

实验与创新实践教育中心

SPI总线工作原理

- ◆ 串行外设接口(Serial Peripheral Interface, SPI) 是由美国摩托罗拉公司最先推出的一种**同步串行传输规范**，也是一种单片机外设芯片串行**扩展接口**，具有**高速**、**全双工**、**接口线少**的特点。
- ◆ 利用**3~4条线**完成两个芯片之间的双工高速通信。2条数据线用于收发数据，1条时钟线用于同步，1条作为从机选择。
- ◆ 总线上允许连接多个设备，在同一时刻只允许**一个主机**操作总线，并且同时只能与**一个从机**通信。

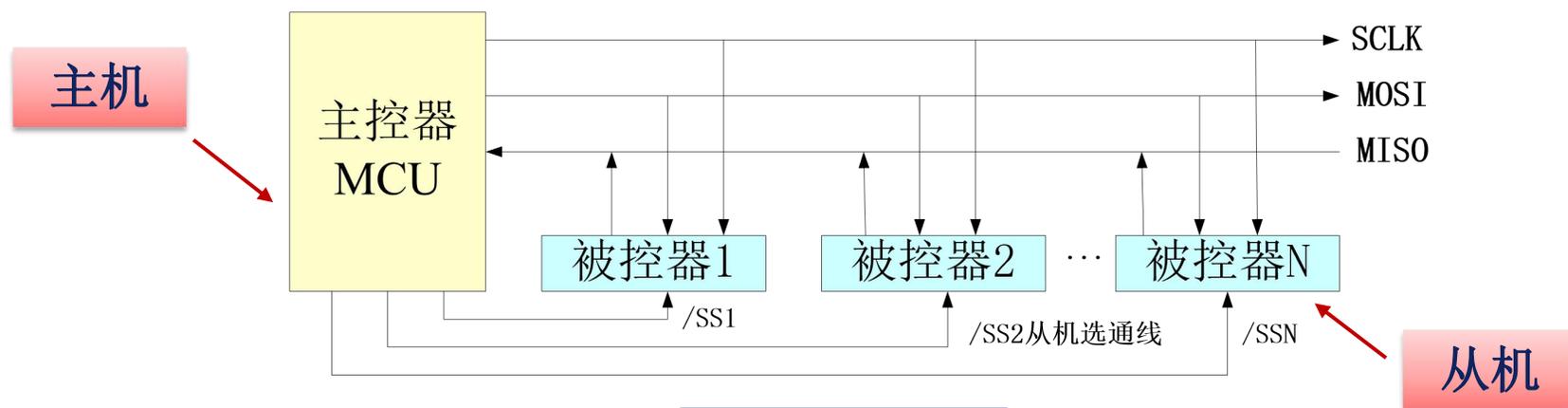
串行/并行通信？

同步/异步？

全双工/半双工/单工？

SPI总线工作原理

- ◆ 一个完整的 SPI 传送周期是 16 位，即两个字节。
- ◆ **传送过程**：首先**主设备**发送命令过去，然后**从设备**根据主设备的命令准备数据，主设备在下一个 8 位时钟周期才把数据读回来。
- ◆ 一个主设备和多个从设备的情况时，每个从设备需要一个单独的片选信号。



SPI典型结构

SPI工作模式

◆ 主机模式：

当器件作为主机时，使用一个IO引脚拉低相应从机的选择引脚(STE)，传输的起始由主机发送数据来启动，时钟(SCLK)信号由主机产生。通过MOSI发送数据，同时通过MISO引脚接收从机发出的数据。

◆ 从机模式：

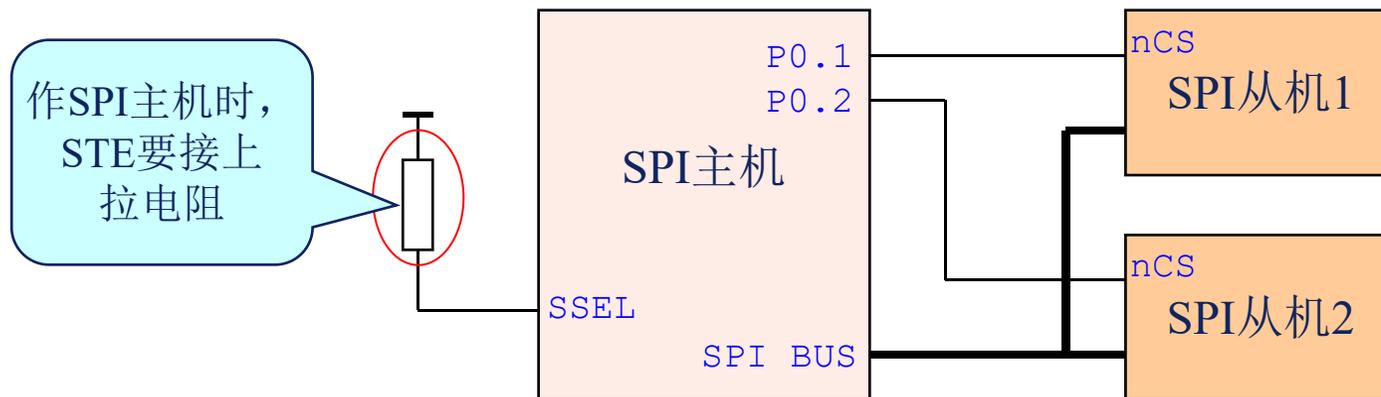
当器件作为从机时，传输在从机选择引脚(STE)被主机拉低后开始，接收主机输出的时钟信号，在读取主机数据的同时通过MISO引脚输出数据。

SPI工作模式

以 4 线SPI为例，通信时需要的4个引脚分别为：

引脚名称	类型	描述
SCLK	输入/输出	串行时钟，用于同步SPI接口间数据传输的时钟信号。该时钟信号总是由主机驱动，从机接收。
STE	输入	从机选择，SPI从机选择信号是一个低有效信号，用于选择进行数据传输的从机。每个从机都有各自特定的从机选择输入信号。
MISO	输入/输出	主设备数据输入，MISO信号是一个单向的信号，它将数据由从机传输到主机。
MOSI	输入/输出	主设备数据输出，MOSI信号是一个单向的信号，它将数据从主机传输到从机。

SPI工作模式



时钟由主设备控制，当主机发送一字节数据（通过主出从入MOSI引脚）的同时，从机返回一字节数据（通过主入从出MISO引脚）。

SPI工作模式

- ◆ **STE**：从机模式发送接收允许控制，控制多主从系统中的多个从机。该引脚不用于3线SPI操作，在4线SPI操作中使多主机共享总线，避免发生冲突。
- ◆ 4线SPI操作**主模式**中，STE的含义如下：
 - SIMO和SCLK被强制进入输入状态
 - SIMO和SCLK正常操作
- ◆ 4线SPI操作**从模式**中，STE的含义如下：
 - 允许从机发送接收数据，SIMO正常操作
 - 禁止从机发送接收数据，SIMO被强制进入输入状态

SPI数据传输

数据传输格式：

通常是高位(MSB)在前，低位(LSB)在后。一些增强型MCU中可以通过软件设置高位在前或低位在前。

以 8 位数据的传输为例，看一下 4 种不同数据传输格式的时序：

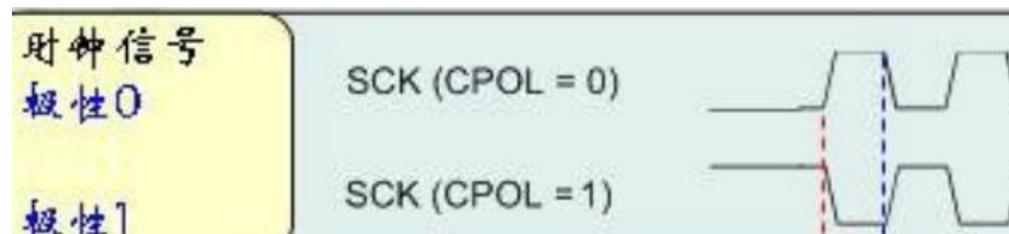
- 1.时钟极性：**表示时钟信号在空闲时是高电平还是低电平。
- 2.时钟相位：**表示数据是在SCLK的上升沿采样还是在SCLK的下降沿采样。

SPI数据传输

时钟极性(CPOL)定义了时钟空闲状态电平，对传输协议没有重大影响。

。

- ◆ CPOL=0: 时钟空闲状态为低电平。
- ◆ CPOL=1: 时钟空闲状态为高电平。



时钟相位(CPHA)定义数据的采样和输出时间。

- ◆ CPHA=0: **Data is changed on the first UCLK edge and captured on the following edge.**
- ◆ CPHA=1: **Data is captured on the first UCLK edge and changed on the following edge.**

SPI数据传输

数据流：稳定时采样，变化时输出前面已经采集好保存的

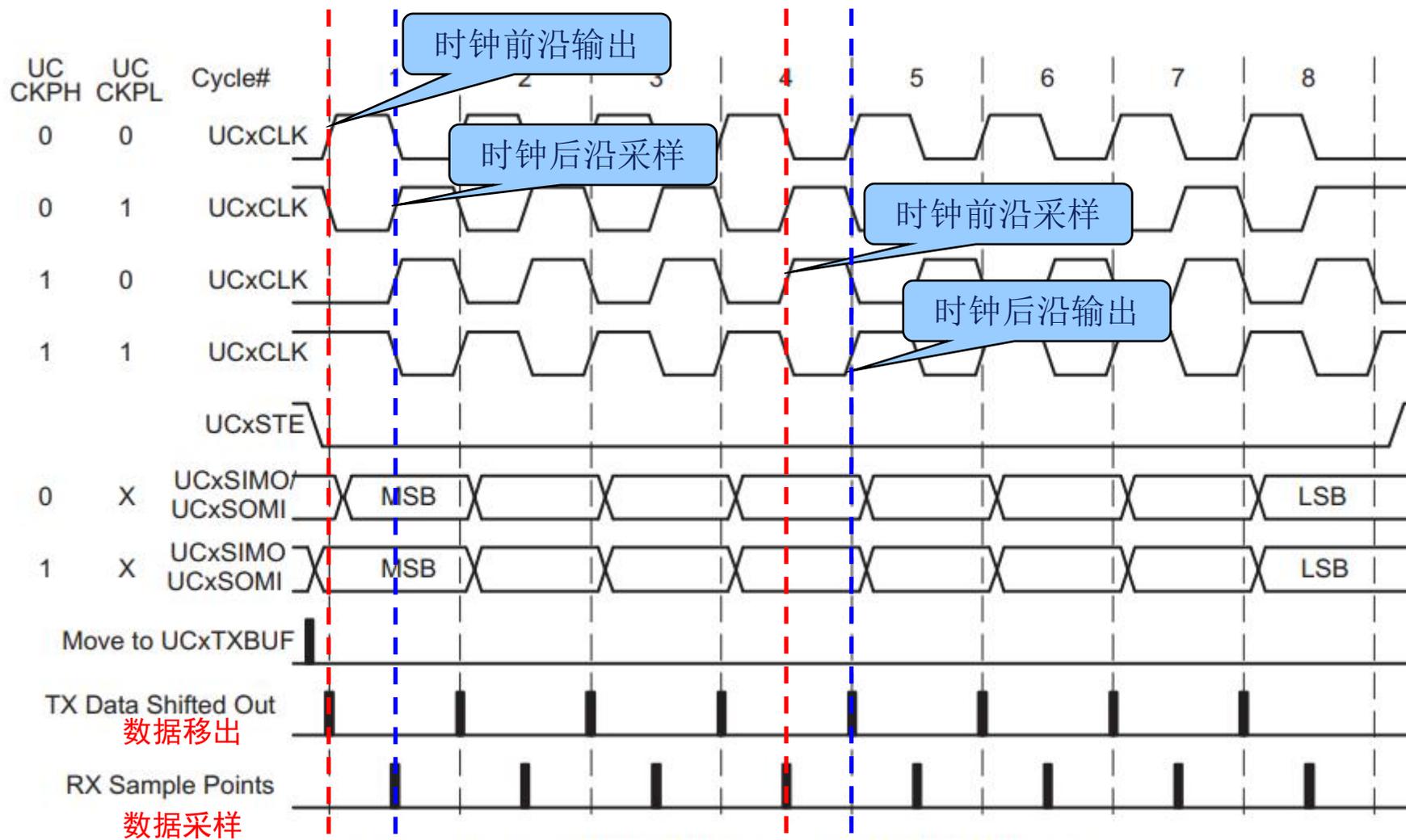


Figure 35-4. USCI SPI Timing With $UCMSB = 1$

SPI数据传输

根据**时钟极性**(CPOL)及**相位**(CPHA)不同可以组合成4种工作模式：SPI0, SPI1, SP2, SP3.

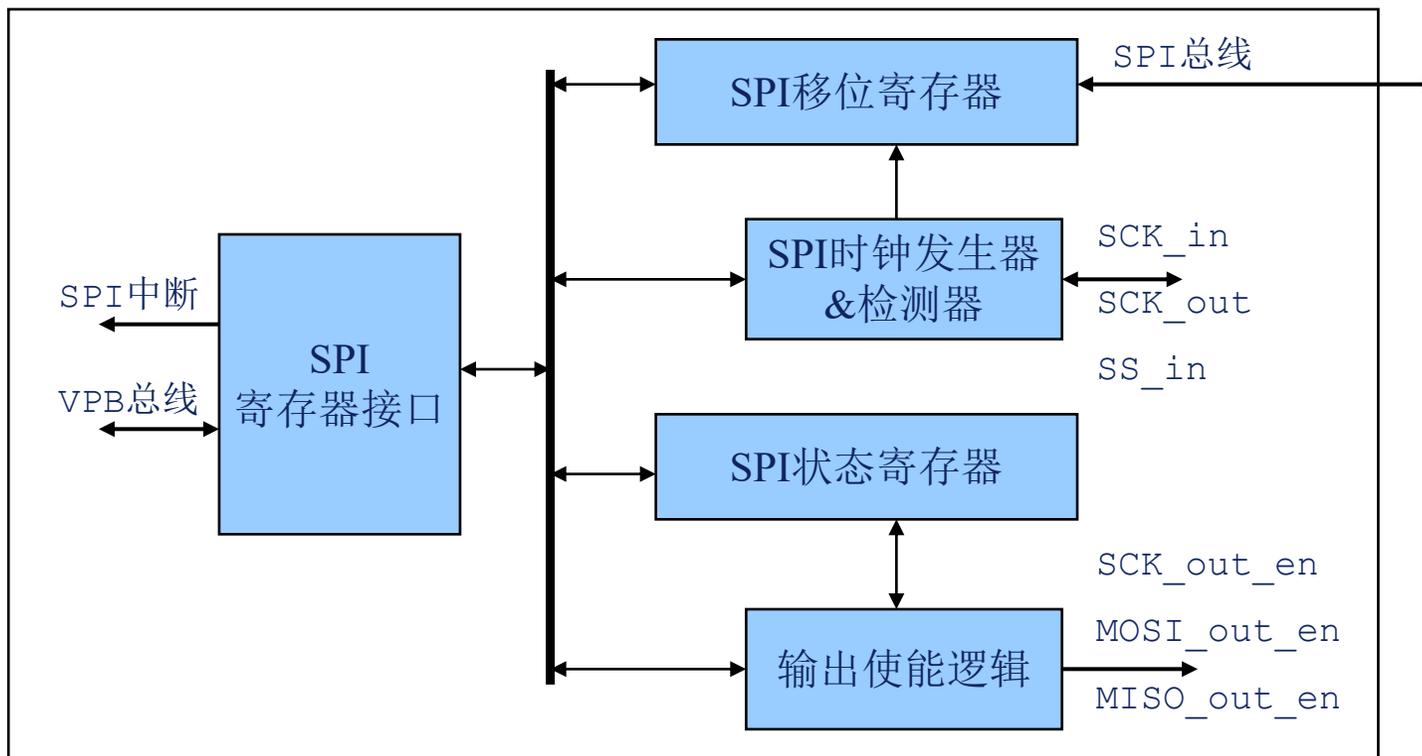
(1) SPI0: CPOL=0,CPHA=0

(2) SPI1: CPOL=0,CPHA=1

(3) SPI2: CPOL=1,CPHA=0

(4) SPI3: CPOL=1,CPHA=1

SPI接口内部结构



SPI特点

优点：

- (1) 接口简单，利于硬件设计与实现。
- (2) 时钟速度快，且没有系统开销。
- (3) 相对抗干扰能力强，传输稳定。

缺点：

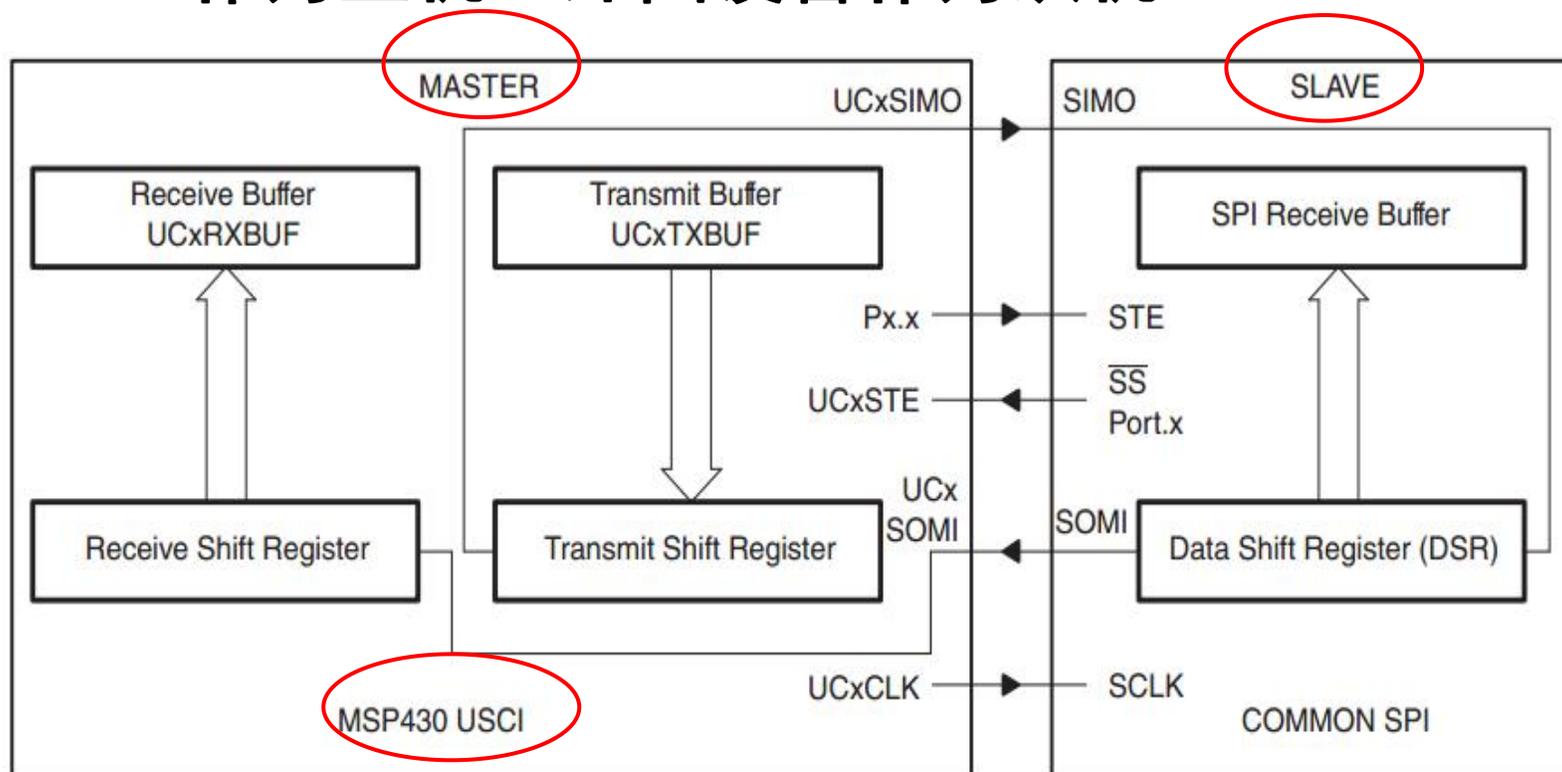
- (1) 缺乏流控制机制，无论主机还是从机均不对消息进行确认，主机无法知道从机是否繁忙，需要软件弥补，增加了软件开发工作量。
- (2) 没有多主机协议，必须采用很复杂的软件和外部逻辑来实现多主机架构。

通用串行通信接口(USCI)模块

- ◆ MSP430单片机通用串行通信接口(USCI)模块支持**多种串行通信模式**。不同的USCI 模块支持不同的模式
- ◆ USCI_Ax 模块支持：
 - UART 模式
 - IrDA 通信的脉冲整形
 - LIN 通信的自动波特率检测
 - **SPI 模式**
- ◆ USCI_Bx 模块支持：
 - I2C 模式
 - **SPI 模式**

通用串行通信接口(USCI)模块

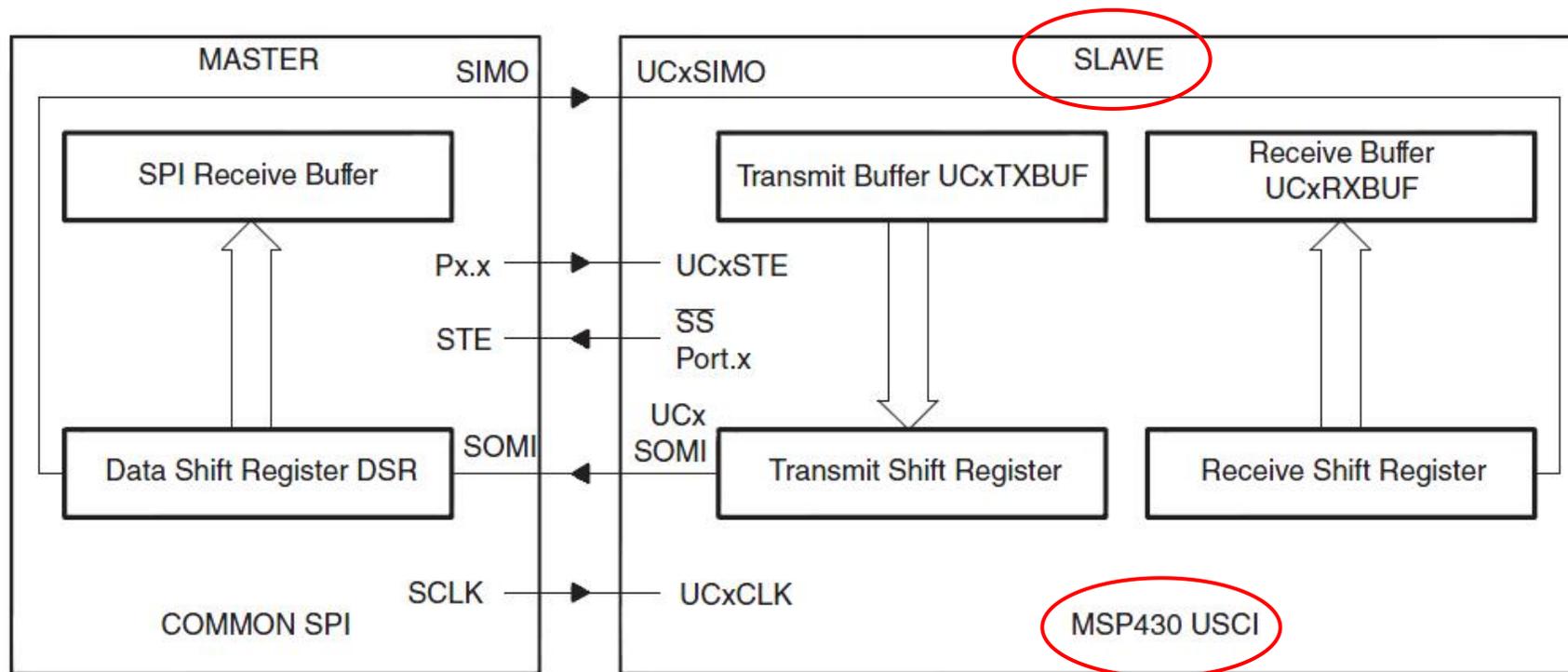
MSP430 USCI作为主机、外围设备作为从机



主模式

通用串行通信接口(USCI)模块

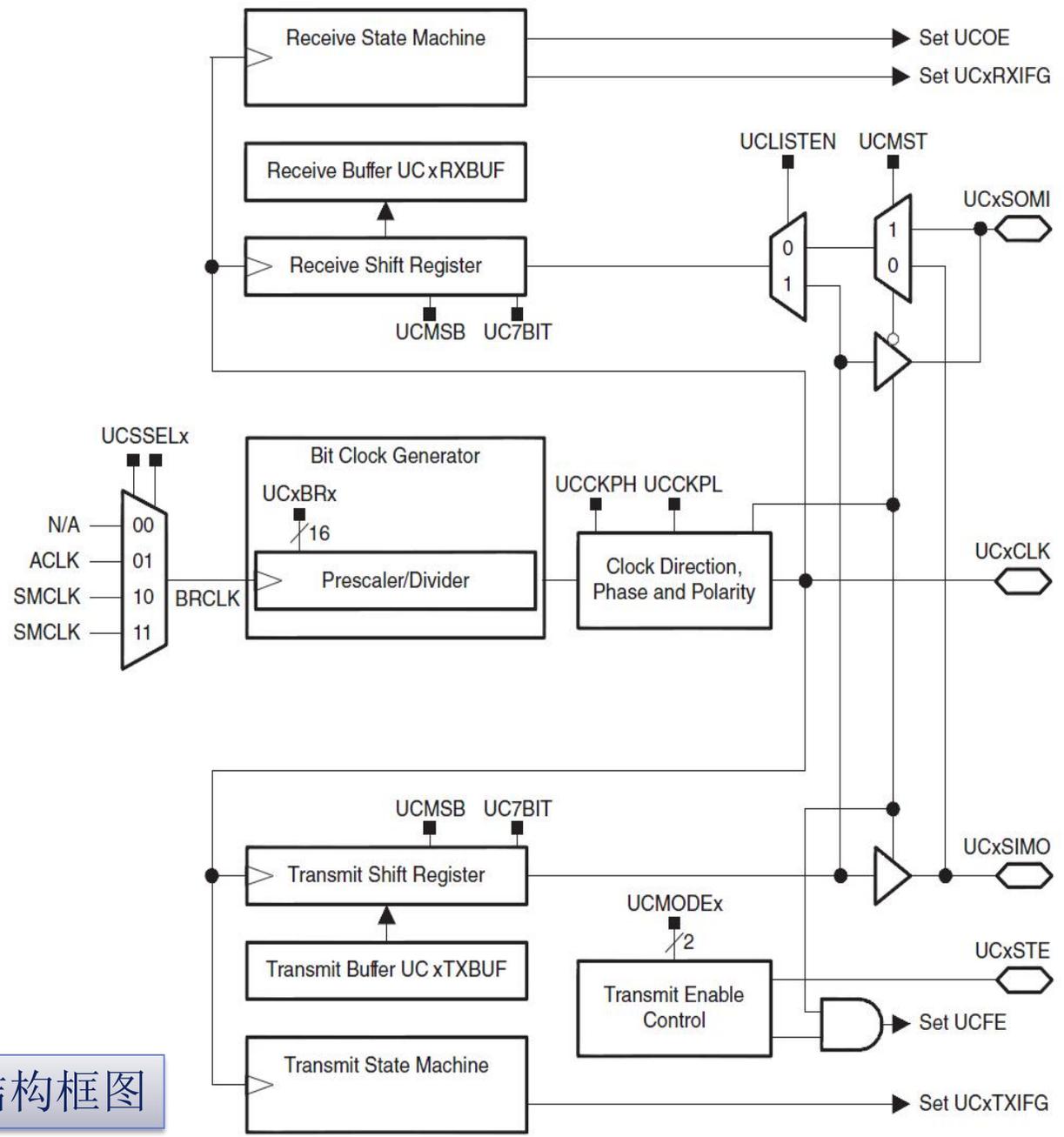
外围设备作为主机，MSP430 USCI作为从机



从模式

图

SPI接口结构框图



SPI模式相关寄存器

USCI_Ax和USCI_Bx都有SPI模块，下面以USCI_Bx为例，介绍相关寄存器

名称	描述	访问	复位值	寄存器访问
UCBxCTLW0	USCI_Bx控制字0	读/写	0001h	字
UCBxBRW	USCI_Bx波特率控制字	读/写	0000h	字
UCBxMCTL	USCI_Bx调制器控制			
UCBxSTAT	USCI_Bx状态寄存器	读/写	00h	字节
UCBxRXBUF	USCI_Bx接收缓存	读/写	00h	字节
UCBxTXBUF	USCI_Bx发送缓存	读/写	00h	字节
UCBxI2COA	USCI_Bx I2C本机地址	读/写	0000h	字
UCBxI2CSA	USCI_Bx I2C从机地址	读/写	0000h	字
UCBxICTL	USCI_Bx中断控制	读/写	0200h	字
UCBxIE	USCI_Bx中断使能	读/写	00h	字节
UCBxIFG	USCI_Bx中断标志	读/写	02h	字节
UCBxIV	USCI_Bx中断向量	读	0000h	字

课堂实验5.1

内容：

结合电子纸屏幕用户手册，参考电子纸屏幕的源代码(华清科仪资料包中)，自行设计显示内容。

要求：

- (1) 显示字符。
- (2) 显示图片。
- (3) 利用GPIO中断，实现1个按键控制显示内容，字符换行或者切换图片。

主要内容

- ◆ 串行通信接口SPI模式
- ⇒ ◆ 展馆灯光综合调试
- ◆ 项目制作准备

该系统主要用于展览馆等需要解说员解说，且需要调节光线以达到最佳演示效果的场合。系统检测到外界声音后打开灯光，系统根据周边环境光的情况自动调整LED灯的亮度，以达到最佳展示效果。如果一定时间内都没有检测到声音信号，则自动关闭LED灯，以达到节能目的。

从中提取关键信息：

- 1.能感知声音信号
- 2.能感知光强信号
- 3.能调节LED灯亮度
- 4.能自动关闭LED灯

声控灯+自动调节亮度

展馆灯光控制

◆ 参考课上实验4.3（回顾）：

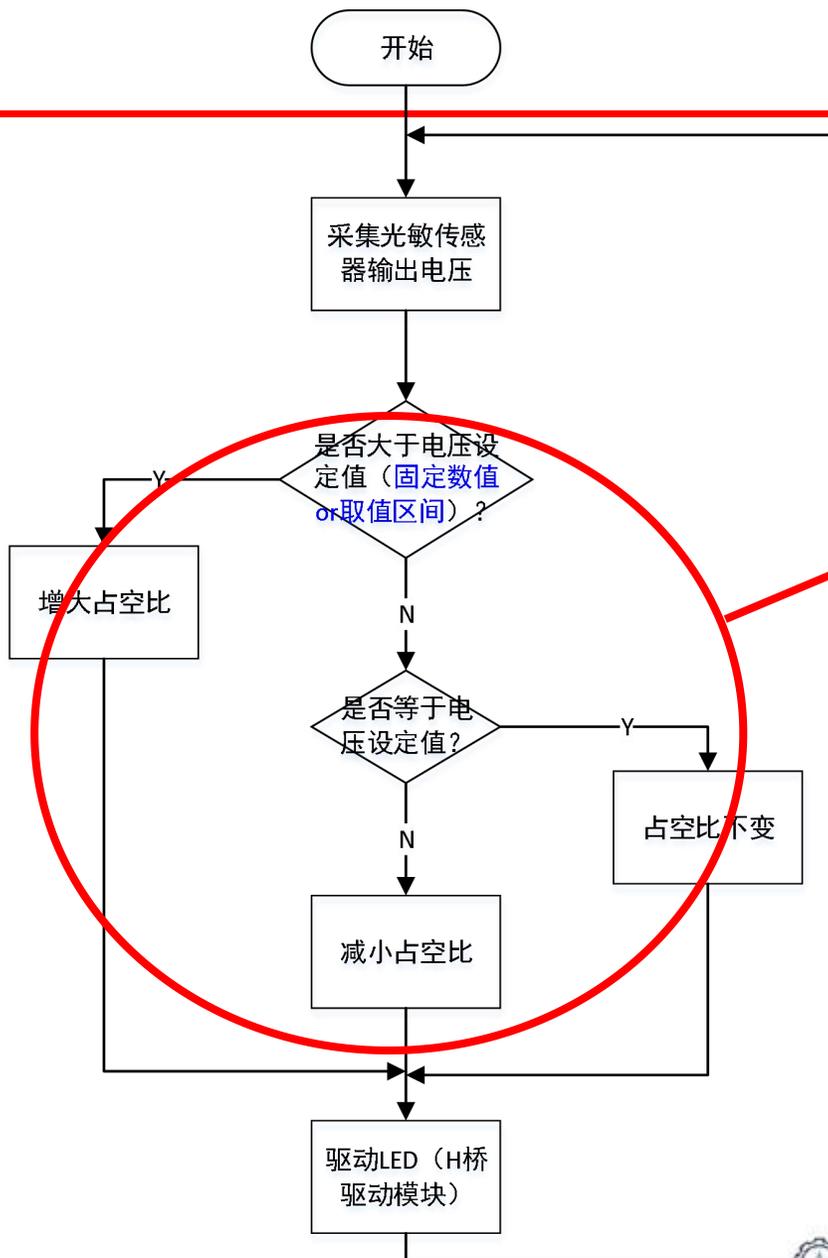
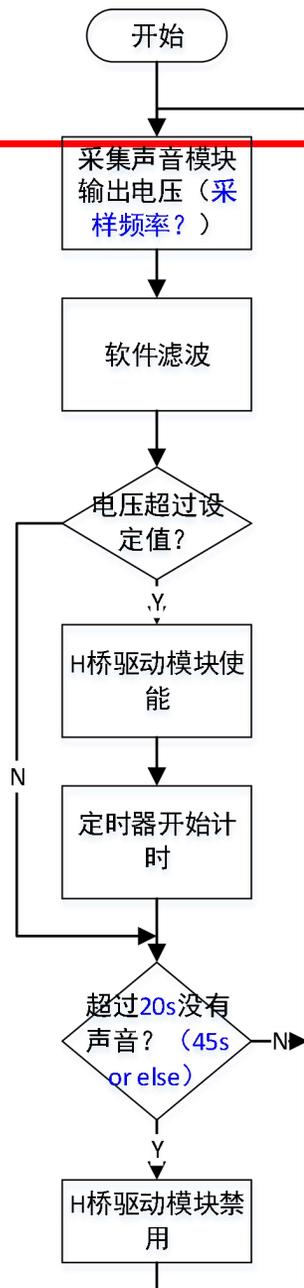
通过AD同时采集声音信号和环境光强信号：

- (1) 当有声音时，主板LED灯L1点亮，**没有声音10s后L1熄灭。**
- (2) 通过口袋板LED灯L1~L6显示环境光强，光越强，LED亮灯数量越多。
- (3) 外界环境光强的处理作为一个接口函数，用来控制PWM信号的占空比，参看定时器A的PWM输出信号呼吸灯实验任务，实现根据外界光强变化，进行主板LED1灯的明暗调节任务；

展馆灯光控制软件

- 概要设计
- 详细设计**
- 编码
- 代码测试

声控子程序



PID控制?

调光子程序

实现展馆灯光控制

课上实验5.2:

尝试参考课上实验4.3要求，调试完成展馆灯光控制系统，可加入屏幕显示增加人机对话功能。

- ◆ 课上完成则找老师登记，完不成不做要求，无需课下补检查。
- ◆ 调试过程注意事项（后页）

展馆灯光控制——AD采集初始化接口

```
void InitAD()
```

```
{
```

```
    ADC12CTL0 |= ADC12MSC;           //自动循环采样转换
```

```
    ADC12CTL0 |= ADC12ON;           //启动ADC12模块
```

```
    ADC12CTL1 |= ADC12CONSEQ_3;     //选择序列通道多次循环采样转换
```

```
    ADC12CTL1 |= ADC12SHP;         //采样保持模式
```

```
    ADC12CTL1 |= ADC12CSTARTADD_0;
```

```
    ADC12MCTL0 |= ADC12INCH_1;     //通道选择
```

```
    ADC12MCTL1 |= ADC12INCH_2+ADC12EOS;
```

```
    ADC12CTL0 |= ADC12ENC;
```

```
}
```

MEM0 存储P6.1的采样数据
MEM1 存储P6.2的采样数据

展馆灯光控制——ADC软件滤波接口（麦克风通道为例）

```
unsigned int micFilter( )
{
    char count,i,j;
    unsigned int value_buf[N_POINT];
    unsigned int temp=0;
    int sum=0;
    for (count=0;count<N_POINT;count++)
    {
        value_buf[count] = GetADmic();
    }
    for (j=0;j<N_POINT-1;j++)
    {
        for (i=0;i<N_POINT-j-1;i++)
        {
            if ( value_buf[i]>value_buf[i+1] )
            {
                temp = value_buf[i];
                value_buf[i] = value_buf[i+1];
                value_buf[i+1] = temp;
            }
        }
    }
}
```

```
for(count=1;count<N_POINT-1;count++)
{
    sum += value_buf[count];
}
//如果为2的n次方等分，则可以用>>n的减少计算量 a=a*8;
//可以为 a=a<<3; b=b/8; 可以为b=b>>3;
return (unsigned int)(sum/(N_POINT-2));
}
```

```
#define N_POINT    13
unsigned int GetADmic()
{
    unsigned int temp = 0;//设置变量

    ADC12CTL0 |= ADC12SC; //开始采样转换
    temp= ADC12MEM1; //把结果赋给变量

    return temp;
}
```

明确采样数据存储位置

展馆灯光控制——时钟初始化接口

```
void ClockInit()
```

```
{//最终MCLK:16MHz, SMCLK:8MHz, ACLK:32KHz
```

```
    UCSCTL6 &= ~XT1OFF;           //启动XT1  
    P5SEL |= BIT2 + BIT3;         //XT2引脚功能选择  
    UCSCTL6 &= ~XT2OFF;           //打开XT2  
    __bis_SR_register(SCG0);  
    UCSCTL0 = DCO0+DCO1+DCO2+DCO3+DCO4;  
    UCSCTL1 = DCORSEL_4;          //DCO频率范围在28.2MHz以下  
    UCSCTL2 = FLLD_5 + 1;         //D=16, N=1
```

```
    //n=8,FLLREFCLK时钟源为XT2CLK;  
    //DCOCLK=D*(N+1)*(FLLREFCLK/n);DCOCLKDIV=(N+1)*(FLLREFCLK/n);  
    UCSCTL3 = SELREF_5 + FLLREFDIV_3;  
    //ACLK的时钟源为DCOCLKDIV,MCLK\SMCLK的时钟源为DCOCLK  
    UCSCTL4 = SELA_4 + SELS_3 + SELM_3;  
    //ACLK由DCOCLKDIV的32分频得到, SMCLK由DCOCLK的2分频得到  
    UCSCTL5 = DIVA_5 + DIVS_1;
```

```
}
```

在定时器中断计时实验中，SMCLK使用的是默认1MHz，原有代码计时10s，在调用此接口后则变成原来时间的1/8，要注意调整代码，保证计时时间。

展馆灯光控制——其他注意事项

- ◆ IO初始化要配合**定时器的初始化**和ADC的初始化来完成；
- ◆ 定时器可能用到两个TimerA，注意初始化语句的完整和正确
 - **中断计时10s**（或其他时间）使用，初始化语句要注意**时钟**的选择
 - **TimerA2**输出**PWM**信号使用，初始化注意**CCRn**的选取，与硬件引脚相关
- ◆ 在**调节LED亮度**时，有**乘除计算**，**数据类型的转换**要特别注意，可用临时变量进行过渡操作，推荐计算过程中使用**双精度**类型

主要内容

- ◆ 串行通信接口SPI模式
- ◆ 展馆灯光综合调试
- ⇒ ◆ 项目制作准备

谢谢大家