

单片机应用技术

项目十一

SPI 串行总线应用

【知识目标】

- 掌握 SPI 串行通信协议
- 了解 ATmega16 单片机 SPI 串行通信接口结构
- 了解与 SPI 串行通信有关的寄存器的功能
- 了解 TLC5615D/A 芯片

【能力目标】

- 掌握 ATmega16 单片机的 SPI 串行通信接口相关寄存器的配置方法
- 掌握 TLC5615D/A 芯片的使用方法
- 掌握简单的单片机 SPI 串行通信总线系统程序的编写、调试方法

【任务 1】 项目知识点学习

❖ 一、 SPI 总线通信协议概述

SPI(Serial Peripheral Interface-- 串行外设接口) 总线系统是一种同步外设接口，允许 MCU 与各种外围设备以串行方式进行通信和数据交换。外围设备包括 FLASHRAM、A/D 转换器、网络控制器、MCU 等。一般使用 4 线制：串行时钟线 (SCK)、主机输入 / 从机输出数据线 MISO、主机输出 / 从机输入数据线 MOSI 和低电平有效的从机选择线 SS。

【任务 1】 项目知识点学习

❖ 二、 ATmega16 单片机 SPI 接口控制与数据传输过程

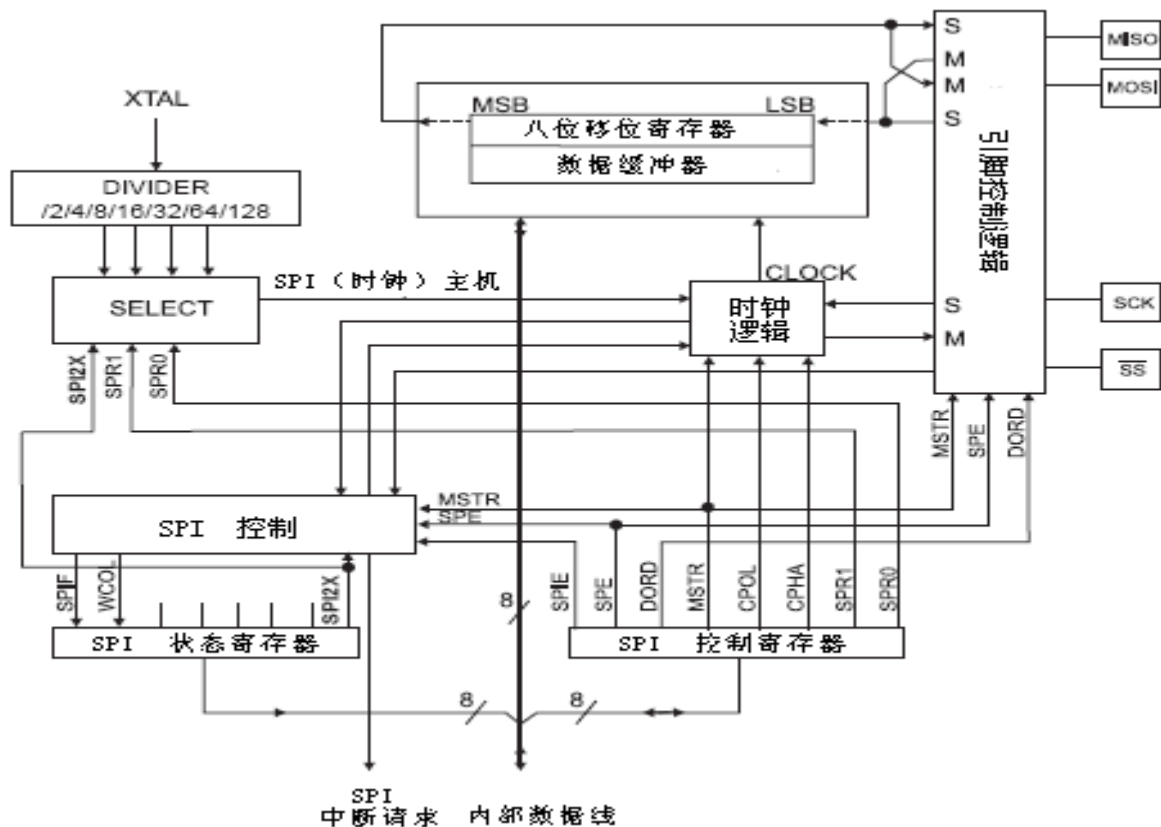
1) 控制与传输过程

如下图所示为 SPI 数据传输系统的结构方框图。SPI 的数据传输系统由主机和从机两个部分构成，主要由主、从机双方的两个移位寄存器和主机 SPI 时钟发生器组成，主机为 SPI 数据传输的控制方。由 SPI 的主机将 SS 输出线拉低，作为同步数据传输的初始化信号，通知从机进入传输状态。然后主机启动时钟发生器，产生同步时钟信号 SCK；预先将在两个移位寄存器中的数据在 SCK 的驱动下进行循环移位操作，实现了主 - 从之间的数据交换。主机的数据由 MOSI（主机输出 - 从机输入）进入从机，而同时从机的数据 MISO（主机输入 - 从机输出）进入主机。数据传送完成，主机将 SS 线拉高，表示传输结束。

【任务 1】 项目知识点学习

❖ 二、 ATmega16 单片机 SPI 接口控制与数据传输过程

1) 控制与传输过程



【任务 1】 项目知识点学习

❖ 二、 ATmega16 单片机 SPI 接口控制与数据传输过程

1) 控制与传输过程

SPI 接口的设置可分为主机和从机两种模式。当 SPI 接口使能时，MOSI、MISO、SCK 和 SS 引脚的控制和数据方向如下表

引脚	方向（主 SPI）	方向（从 SPI）
MOSI	用户定义	输入
MISO	输入	用户定义
SCK	用户定义	输入
SS	用户定义	输入

【任务 1】 项目知识点学习

❖ 二、 ATmega16 单片机 SPI 接口控制与数据传输过程

2) SPI 初始化及数据传送程序示例

下面将以 ATmega16 单片机为例说明如何将 SPI 设置为主机，以及如何进行简单的数据传送。**MOSI** 对应 ATmega16 单片机 **PB5** 引脚，**MISO** 对应 ATmega16 单片机 **PB6** 引脚，**SCK** 对应 ATmega16 单片机 **PB7** 管脚，**SS** 对应 ATmega16 单片机 **PB4** 管脚。

【任务 1】 项目知识点学习

❖ 二、 ATmega16 单片机 SPI 接口控制与数据传输过程

2) SPI 初始化及数据传送程序示例

(1) 设置 SPI 为主机并进行简单的数据发送：

```
void spi_masterset(void)
{
    /* 设置 MOSI 和 SCK 为输出，其他为输入 */
    DDRB = (1<<DDB5) | (1<<DDB7);
    /* 使能 SPI 主机模式，设置时钟速率为 fck/16 */
    SPCR = (1<<SPE) | (1<<MSTR) | (1<<SPR0);
}

void spi_mastertransmit(char data)
{
    /* 启动数据传输 */
    SPDR = data;
    /* 等待传输结束 */
    while(!(SPSR & (1<<SPIF)));
}
```

【任务 1】 项目知识点学习

❖ 二、 ATmega16 单片机 SPI 接口控制与数据传输过程

2) SPI 初始化及数据传送程序示例

(2) 设置 SPI 为从机并进行简单的数据接收。

```
void spi_slaveset(void)
{
    /* 设置 MISO 为输出, 其他为输入 */
    DDRB = (1<<DDB6);
    /* 使能 SPI */
    SPCR = (1<<SPE);
}
char spi_slavereceive(void)
{
    /* 等待接收结束 */
    while(!(SPSR & (1<<SPIF)));
    /* 返回数据 */
    return SPDR;
}
```

【任务 1】 项目知识点学习

❖ 二、 ATmega16 单片机 SPI 接口控制与数据传输过程

3) SS 引脚的功能

· 从机方式

当 SPI 配置为从机时，从机选择引脚 **SS** 总是为**输入**。**SS 为低将激活 SPI 接口**，**MISO** 成为**输出**（用户必须进行相应的端口配置）引脚，**其他引脚**成为**输入引脚**。

【任务 1】 项目知识点学习

❖ 二、 ATmega16 单片机 SPI 接口控制与数据传输过程

3) SS 引脚的功能

· 主机方式

当 SPI 被配置为主机时（寄存器 **SPCR** 的 **MSTR** 位置 “1” ），用户可以决定 SS 引脚方向。如果 SS 引脚被设为输出，该引脚将作为通用输出口，不影响 SPI 系统，通常用于驱动从机的 SS 引脚。

【任务 1】 项目知识点学习

❖ 三、 ATmega16 SPI 接口相关的寄存器简单介绍

1) SPI 控制寄存器 - SPCR

位	7	6	5	4	3	2	1	0	
	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初始值	0	0	0	0	0	0	0	0	

- **Bit 7 – SPIE:** 使能 SPI 中断
置位后，只要 SPSR 寄存器的 SPIF 和 SREG 寄存器的全局中断使能位置位，就会引发 SPI 中断。
- **Bit 6 – SPE:** 使能 SPI
SPE 置位将使能 SPI。进行任何 SPI 操作之前必须置位 SPE。
- **Bit 5 – DORD:** 数据次序
DORD 置位时数据的 LSB 首先发送；否则数据的 MSB 首先发送。

【任务 1】 项目知识点学习

❖ 三、 ATmega16 SPI 接口相关的寄存器简单介绍

1) SPI 控制寄存器 - SPCR

位	7	6	5	4	3	2	1	0	
	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初始值	0	0	0	0	0	0	0	0	

- **Bit 4 – MSTR:** 主 / 从选择

MSTR 置位时选择主机模式，否则为从机。如果 MSTR 为 "1"，SS 配置为输入，但被拉低，则 MSTR 被清零，寄存器 SPSR 的 SPIF 置位。用户必须重新设置 MSTR 进入主机模式。

- **Bit 3 – CPOL:** 时钟极性

CPOL 置位表示空闲时 SCK 为高电平；否则空闲时 SCK 为低电平。

- **Bit 2 – CPHA:** 时钟相位

CPHA 决定数据是在 SCK 的起始沿采样还是在 SCK 的结束沿采样。

【任务 1】 项目知识点学习

❖ 三、 ATmega16 SPI 接口相关的寄存器简单介绍

1) SPI 控制寄存器 - SPCR

位	7	6	5	4	3	2	1	0	
	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初始值	0	0	0	0	0	0	0	0	

- Bits 1, 0 – SPR1, SPR0: SPI 时钟速率选择 1 与 0

确定主机的 SCK 速率。SPR1 和 SPR0 对从机模式没有影响。SCK 和振荡器频率 f_{OSC} 之间的关系下表所示。

SPI2X	SPR1	SPR0	SCK 频率 (MHz)	
0	0	0	fOSC/4	
0	0	1	fOSC/16	
0	1	0	fOSC/64	
0	1	1	fOSC/128	
1	0	0	fOSC/2	
1	0	1	fOSC/8	
	1	1	0	fOSC/32
1	1		1	fOSC/64

【任务 1】 项目知识点学习

❖ 三、 ATmega16 SPI 接口相关的寄存器简单介绍

2) SPI 的状态寄存器 - SPSR

位	7	6	5	4	3	2	1	0	
	SPIF	WCOL	-	-	-	-	-	SPI2X	SPSR
读/写	R	R	R	R	R	R	R	R/W	
初始值	0	0	0	0	0	0	0	0	

- **Bit 7 – SPIF: SPI 中断标志**

串行发送结束后，SPIF 置位。若此时寄存器 SPCR 的 SPIE 和全局中断使能位置位，SPI 中断即产生。如果 SPI 为主机，SS 配置为输入，且被拉低，SPIF 也将置位。进入中断服务程序后 SPIF 自动清零。或者可以通过先读 SPSR，紧接着访问 SPDR 来对 SPIF 清零。

- **Bit 6 – WCOL: 写碰撞标志**

在发送当中对 SPI 数据寄存器 SPDR 写数据将置位 WCOL。WCOL 可以通过先读 SPSR，紧接着访问 SPDR 来清零。

【任务 1】 项目知识点学习

❖ 三、 ATmega16 SPI 接口相关的寄存器简单介绍

2) SPI 的状态寄存器 - SPSR

位	7	6	5	4	3	2	1	0	
	SPIF	WCOL	—	—	—	—	—	SPI2X	SPSR
读 / 写	R	R	R	R	R	R	R	R/W	
初始值	0	0	0	0	0	0	0	0	

- Bit 5.1 – Res: 保留

保留位，读操作返回值为零。

- Bit 0 – SPI2X: SPI 倍速

置位后 SPI 的速度加倍。若为主机，则 SCK 频率可达 CPU 频率的一半。若为从机，只能保证 $f_{osc} / 4$ 。

【任务 1】 项目知识点学习

❖ 三、 ATmega16 SPI 接口相关的寄存器简单介绍

3) SPI 数据寄存器 - SPDR

位	7	6	5	4	3	2	1	0	
	MSB							LSB	SPDR
读 / 写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初始值	X	X	X	X	X	X	X	X	

- **SPI 数据寄存器**为读 / 写寄存器，用来在寄存器文件和 SPI 移位寄存器之间传输数据。写寄存器将启动数据传输，读寄存器将读取寄存器的接收缓冲器。

【任务 1】 项目知识点学习

❖ 四、DA 转换芯片 TLC5615 的相关知识

1) TLC5615 的工作特性

TLC5615 是带有 **3 线**串行接口且具有缓冲输入的 **10 位 DAC**，输出可达 **2 倍 Ref** 的变化范围。其特点如下：

- 5V 单电源工作。
- 3 线制串行接口。
- 高阻抗基准输入。
- 电压可达基准电压两倍。
- 内部复位。

【任务 1】 项目知识点学习

❖ 四、DA 转换芯片 TLC5615 的相关知识

2) TLC5615 的引脚及功能说明

TLC5615 的引脚如图所示，各引脚功能如下：

DIN：串行数据输入端。

SCLK：串行时钟输入端。

CS：片选信号。

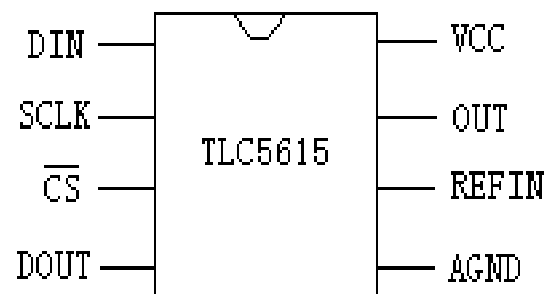
DOUT：串行数据输出端，用于级联。

AGND：模拟地。

REFIN：基准电压输入。

OUT：DAC 模拟电压输出端。

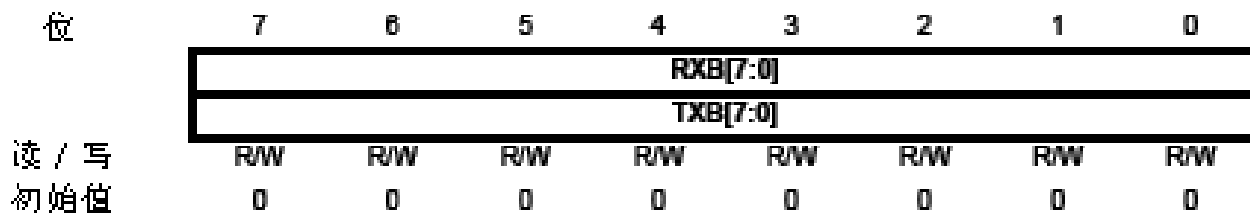
VDD：电源端。



【任务 1】 项目知识点学习

1) 数据缓冲器 UDR

UDR 数据缓冲器的格式图如图 1.7 所示：



ATmega16 单片机 USART 发送数据缓冲寄存器和 USART 接收数据缓冲寄存器**共享相同的 I/O 地址**，称为 USART 数据寄存器或 UDR。将数据写入 UDR 时实际操作的是发送数据缓冲寄存器 (TXB)，读 UDR 时实际返回的是接收数据缓冲寄存器 (RXB) 的内容。

只有当 UCSRA 寄存器的 **UDRE 标志置位**后才可以对发送缓冲器进行写操作。如果 UDRE 没有置位，那么写入 UDR 的数据会被 USART 发送器忽略。当数据写入发送缓冲器后，若移位寄存器为空，发送器将把数据加载到发送移位寄存器。然后数据串行地从 TxD 引脚输出。

【任务 1】 项目知识点学习

2) 控制状态寄存器 UCSRA、UCSRB、UCSRC

(1) 控制状态寄存器 UCSRA

位	7	6	5	4	3	2	1	0	
	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	UCSRA
读 / 写	R	R/W	R	R	R	R	R/W	R/W	
初始值	0	0	1	0	0	0	0	0	

RXC: USART 接收结束

接收缓冲器中有未读出的数据时 RXC 置位，否则清零。接收器禁止时，接收缓冲器被刷新，导致 RXC 清零。RXC 标志可用来产生接收结束中断（见对 RXCIE 位的描述）。

TXC: USART 发送结束

发送移位缓冲器中的数据被送出，且当发送缓冲器 (UDR) 为空时 TXC 置位。执行发送结束中断时 TXC 标志自动清零，也可以通过写 1 进行清除操作。TXC 标志可用来产生发送结束中断（见对 TXCIE 位的描述）。

【任务 1】 项目知识点学习

位	7	6	5	4	3	2	1	0	
	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	UCSRA
读 / 写	R	R/W	R	R	R	R	R/W	R/W	
初始值	0	0	1	0	0	0	0	0	

FE: 帧错误

如果接收缓冲器接收到的下一个字符有帧错误，**即接收缓冲器中的下一个字符的第一个停止位为 0，那么 FE 置位。**这一位一直有效直到接收缓冲器 (UDR) 被读取。当接收到的停止位为 1 时，FE 标志为 0。对 UCSRA 进行写入时，这一位要写 0。

DOR: 数据溢出

数据溢出时 DOR 置位。**当接收缓冲器满（包含了两个数据），接收移位寄存器又有数据，若此时检测到一个新的起始位，数据溢出就产生了。**这一位一直有效直到接收缓冲器 (UDR) 被读取。对 UCSRA 进行写入时，这一位要写 0。

【任务 1】 项目知识点学习

位	7	6	5	4	3	2	1	0	
读 / 写	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	UCSRA
初始值	0	0	1	0	0	0	0	0	

PE: 奇偶校验错误

当奇偶校验使能 ($UPM1 = 1$)，且接收缓冲器中所接收到的下一个字符有奇偶校验错误时 UPE 置位。这一位一直有效直到接收缓冲器 (UDR) 被读取。对 UCSRA 进行写入时，这一位要写 0。

U2X: 倍速发送

这一位仅对异步操作有影响。使用同步操作时将此位清零。此位置 1 可将波特率分频因子从 16 降到 8，从而有效的将异步通信模式的传输速率加倍。

MPCM: 多处理器通信模式

设置此位将启动多处理器通信模式。MPCM 置位后，USART 接收器接收到的那些不包含地址信息的输入帧都将被忽略。发送器不受 MPCM 设置的影响。

【任务 1】 项目知识点学习

(2) 控制状态寄存器 UCSRB 格式

位	7	6	5	4	3	2	1	0	
	RXCIE TXCIE		UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	UCSRB
读 / 写	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
初始值	0	0	0	0	0	0	0	0	

RXCIE: 接收结束中断使能 *** 向量号 :

置位后使能 RXC 中断。当 RXCIE 为 1，全局中断标志位 SREG 置位，UCSRA 寄存器的 RXC 亦为 1 时可以产生 USART 接收结束中断。

TXCIE: 发送结束中断使能 *** 向量号 :

置位后使能 TXC 中断。当 TXCIE 为 1，全局中断标志位 SREG 置位，UCSRA 寄存器的 TXC 亦为 1 时可以产生 USART 发送结束中断。

【任务 1】 项目知识点学习

位	7	6	5	4	3	2	1	0	
	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	UCSRB
读 / 写	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
初始值	0	0	0	0	0	0	0	0	

UDRIE: USART 数据寄存器空中断使能. 置位后使能 UDRE 中断。当 UDRIE 为 1，全局中断标志位 SREG 置位，UCSRA 寄存器的 UDRE 亦为 1 时可以产生 USART 数据寄存器空中断。 ***** 向量号：**

RXEN: 接收使能. 置位后将启动 USART 接收器。RxD 引脚的通用端口功能被 USART 功能所取代。禁止接收器将刷新接收缓冲器，并使 FE、DOR 及 PE 标志无效。

TXEN: 发送使能置. 位后将启动将启动 USART 发送器。TxD 引脚的通用端口功能被 USART 功能所取代。TXEN 清零后，只有等到所有的数据发送完成后发送器才能够真正禁止，即发送移位寄存器与发送缓冲寄存器中没有要传送的数据。发送器禁止后，TxD 引脚恢复其通用 I/O 功能。

【任务 1】 项目知识点学习

位	7	6	5	4	3	2	1	0	
	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	UCSRB
读 / 写	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
初始值	0	0	0	0	0	0	0	0	

UCSZ2: 字符长度

UCSZ2 与 UCSRC 寄存器的 UCSZ1:0 结合在一起可以设置数据帧所包含的数据位数 (字符长度)。

RXB8: 接收数据位 8

对 9 位串行帧进行操作时，RXB8 是第 9 个数据位。读取 UDR 包含的低位数据之前首先要读取 RXB8。

TXB8: 发送数据位 8

对 9 位串行帧进行操作时，TXB8 是第 9 个数据位。写 UDR 之前首先要对它进行写操作。

【任务 1】 项目知识点学习

(3) 控制状态寄存器 UCSRC

位	7	6	5	4	3	2	1	0	
	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	UCSRC
读 / 写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初始值	1	0	0	0	0	1	1	0	

在 ATmega16 单片机中，UCSRC 寄存器与 UBRRH 寄存器共用相同的 I/O 地址。对控制寄存器 UCSRC 的各位介绍如下：

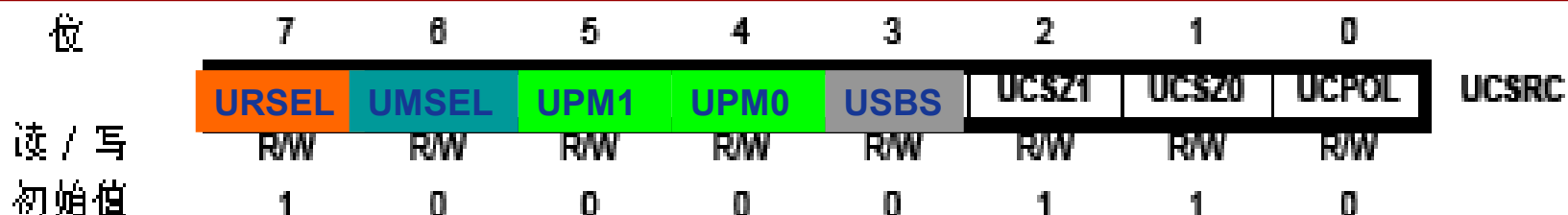
URSEL: 寄存器选择

通过该位选择访问 UCSRC 寄存器或 UBRRH 寄存器。当读 UCSRC 时，该位为 1；当写 UCSRC 时，URSEL 为 1。

UMSEL: USART 模式选择

当 UMSEL 位为 0 时，串行口工作于异步操作模式；当 UMSEL 位为 1 时，串行口工作于同步操作模式。

【任务 1】 项目知识点学习

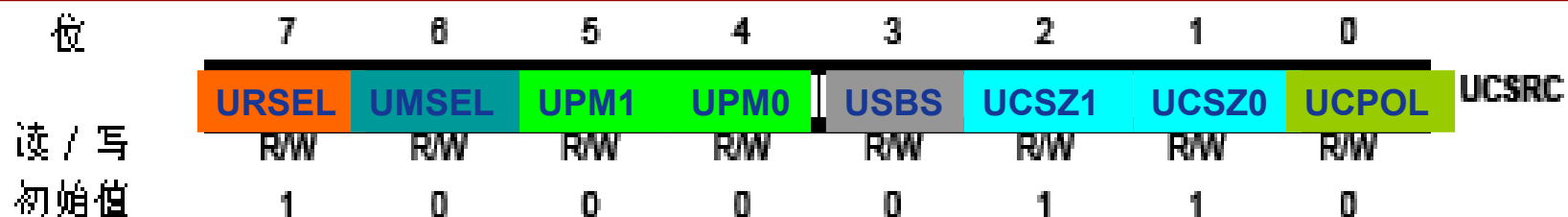


- ❖ **UPM1:0: 奇偶校验模式** . 这两位设置奇偶校验的模式并使能奇偶校验。如果使能了奇偶校验，那么在发送数据，发送器都会自动产生并发送奇偶校验位。对每一个接收到的数据，接收器都会产生一奇偶值，并与接收到的数据进行比较。如果不匹配，那么就产生一个错误标志。ATmega16 单片机串行接口的奇偶校验模式设置如表 1.1 所示。

UPM1	UPM0	奇偶模式
0	0	禁止
0	1	保留
1	0	偶校验
1	1	奇校验

- ❖ **USBS: 停止位选择**通过这一位可以设置停止位的位数。接收器忽略这一位的设置。当 USBS 位为 0 时，停止位位数为 1；当 USBS 位为 1 时，停止位位数为 2。

【任务 1】 项目知识点学习



- ❖ UCSZ1:0: 字符长度 .UCSZ1:0 与 UCSRB 寄存器的 UCSZ2 结合在一起可以设置数据帧包含的数据位数 (字符长度)。其具体设置如表所示：

UCSZ2	UCSZ1	UCSZ0	字符长度
0	0	0	5 位
0	0	1	6 位
0	1	0	7 位
0	1	1	8 位
1	0	0	保留
1	0	1	保留
1	1	0	保留
1	1	1	9 位

- ❖ UCPOL: 时钟极性 . 这一位仅用于同步工作模式。使用异步模式时，将这一位清零。UCPOL 设置了输出数据的改变和输入数据采样，以及同步时钟 XCK 之间的关系。

【任务 1】 项目知识点学习

3) 波特率设置寄存器

波特率寄存器 UBRRL、UBRRH 格式

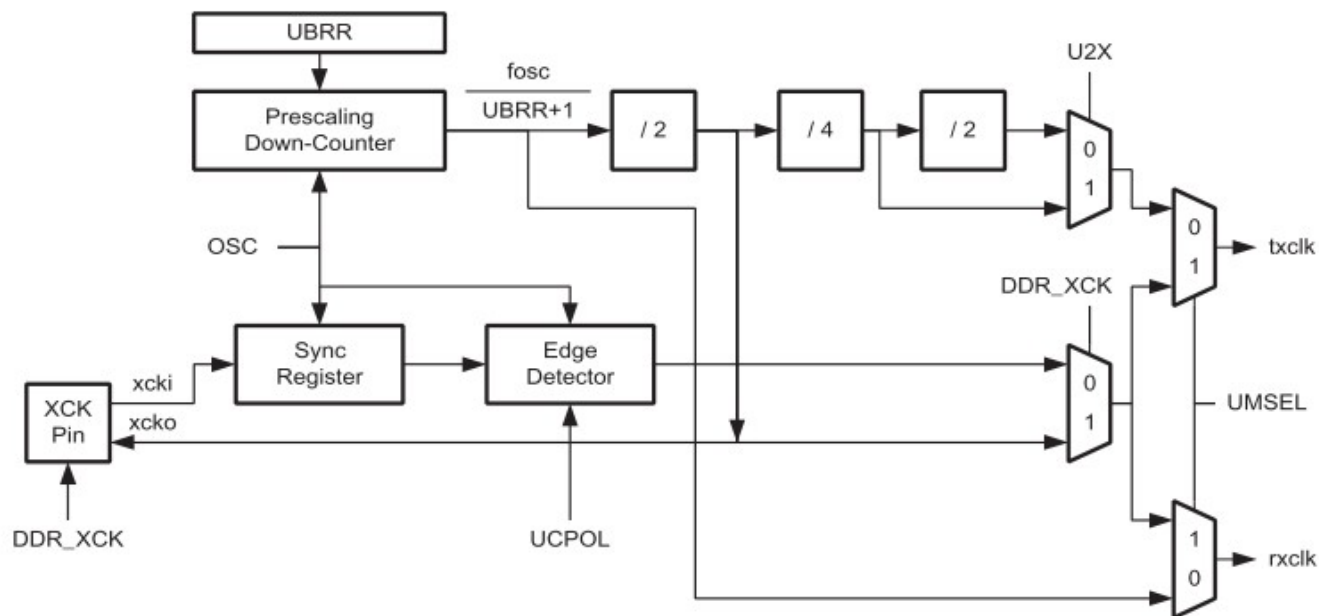
位	15	14	13	12	11	10	9	8			
	URSEL	—	—	—	UBRR 11:8				UBRRH		
	UBRR 7:0										UBRRL
读 / 写	R/W	R	R	R	R/W	R/W	R/W	R/W			
初始值	0	0	0	0	0	0	0	0			
	0	0	0	0	0	0	0	0			

URSEL: 寄存器选择通过该位选择访问 UCSRC 寄存器或 UBRRH 寄存器。当读 UBRRH 时，该位为 0；当写 UBRRH 时，URSEL 为 0。

- ❖ **Bit 14:12 – 保留位** 这些位是为以后的使用而保留的。为了与以后的器件兼容，写 UBRRH 时将这些位清零。
- ❖ **Bit 11:0 – UBRR11:0: USART 波特率寄存器**。这个 12 位的寄存器包含了 USART 的波特率信息。其中 UBRRH 包含了 USART 波特率高 4 位，UBRRL 包含了低 8 位。波特率的改变将造成正在进行的数据传输受到破坏。写 UBRRL 将立即更新波特率分频器。

【任务 1】 项目知识点学习

4) 时钟的产生 - 波特率发生器



❖ 信号说明：

❖ **txclk** 发送器时钟（内部信号） **rxclk** 接收器基础时钟（内部信号）

❖ **xcki** XCK 引脚输入（内部信号），用于同步从机操作；

❖ **xcko** 输出到 XCK 引脚的时钟（内部信号），用于同步主机操作；

❖ **fosc** XTAL 频率（系统时钟）

【任务 2】 TLC5615D/A 转换器应用

❖ 一、任务要求

- 利用 ATmega16 单片机的 SPI 接口输出不同的数字量，通过控制 DA 转换芯片 TLC5615 转换为不同的模拟电压值来实现 LED 小灯的渐亮、渐灭。

【任务 2】 TLC5615D/A 转换器应用

二、硬件设计

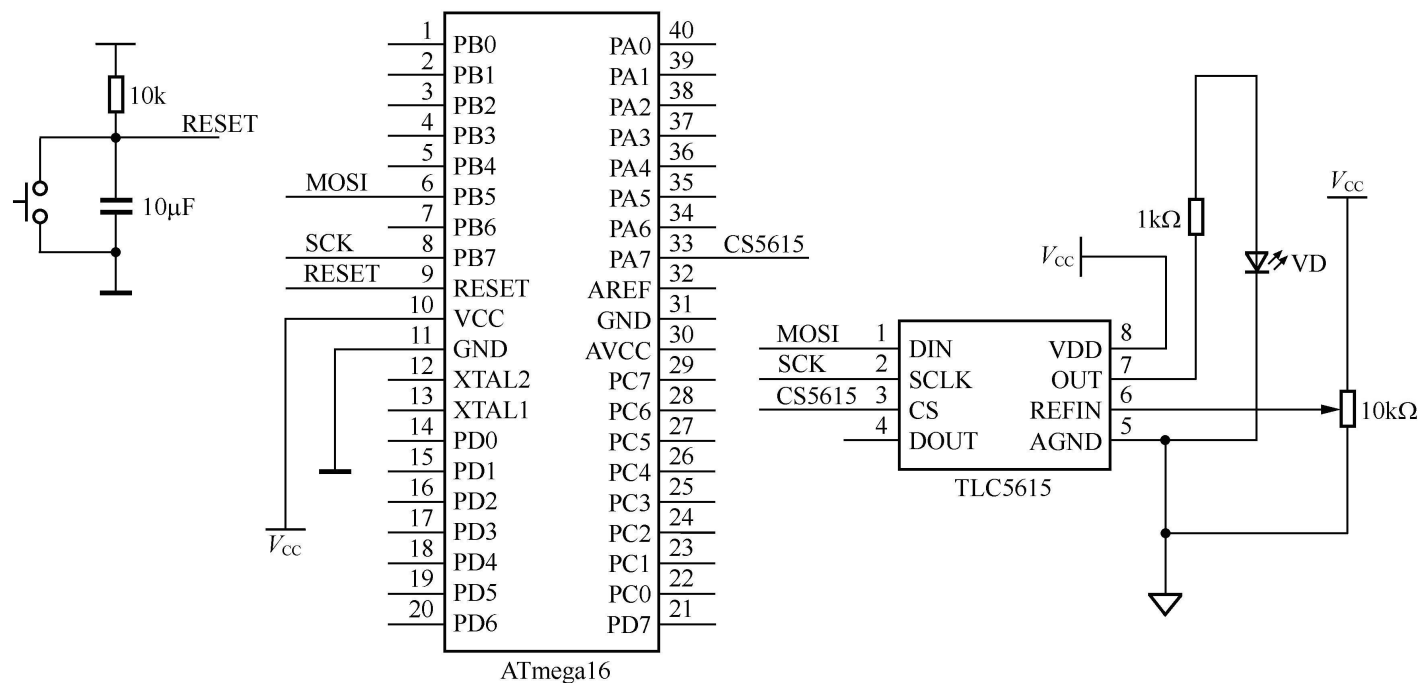


图 10-8 硬件连线示意图

【任务 2】 TLC5615D/A 转换器应用

三、程序设计

ATmega16 单片机为该电路的核心控制器件，通过 SPI 同步串行传输数据的方法，向 DA 转换芯片 TLC5615 送入相关数字量，经转换后输出模拟电压量。D 为一发光二极管，通过限流电阻接于输出端。输出电压随输入数字量的变化而变化，从而实现了小灯的渐亮、渐灭，如图 11-4 所示。R1 为一个电位器，由它改变 TLC5615 DA 转换芯片的参考电压。

【任务 2】 TLC5615D/A 转换器应用

三、程序设计

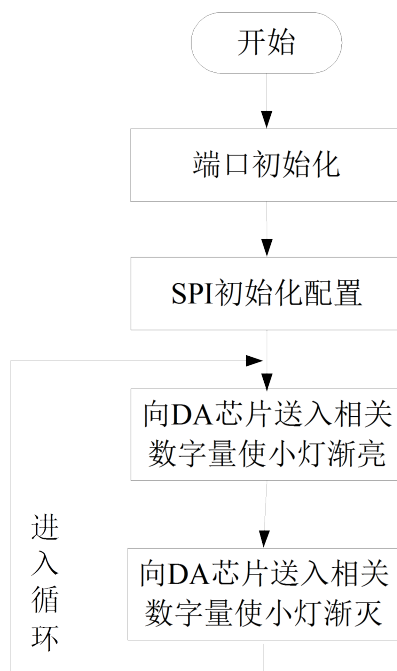


图 11-4 程序流程图

【任务 2】 TLC5615D/A 转换器应用

四、项目实施

1. 根据元器件清单选择合适的元器件。
2. 根据硬件设计原理图，在万能电路板进行元器件布局，并进行焊接工作。
3. 焊接完成后，重复进行线路检查，防止短路、虚接现象。
4. 在 AVR Studio 软件中创建项目，输入源代码并生成 *.hex 文件。
5. 在确认硬件电路正确的前提下，通过 JTAG 仿真器进行程序的下载与硬件在线调试。