ارتباط سریال SPI

SPI یک استاندارد سریال سنکرون می باشد که مخفف Serial Peripheral interface بوده و به وسیله شرکت موتورولا طراحی شده است . این استاندارد به لحاظ پشتیبانی از سرعت بالا نه تنها در کاربردهای اندازه گیری بلکه در مواردی نظیر انتقال حجم بالای اطلاعات و پردازش سیگنال دیجیتال وغیره مورد استفاده واقع می شود. سرعت چند مگابایت بر ثانیه به راحتی توسط SPI قابل دسترسی است و در نتیجه امکان ارسال و دریافت صوت فشرده نشده وجود خواهد داشت .

مزایای SPI :

1- ارتباط دو طرفه سریال (Full Duplex)

2- میکرو هم می تواند به عنوان Slave و هم به صورت Master به کار رود

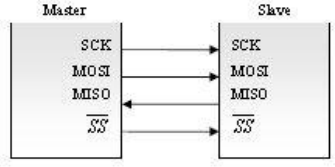
3- ایجاد همزمانی

4- استفاده از مد idle (با انتقال کامل یک بایت داده از مد idle بیدار می شود)

5- انتقال بیت LSB ویا MSB در ابتدا

6- دو برابر کردن سرعت

نحوه ارتباط دو میکرو به صورت SPI در شکل زیر امده است :



شکل 1-10

به هنگام استفاده از SPI باید به نکات زیر توجه کرد :

- وضعیت پایه های MOSI و MISO از نظر ورودی و خروجی باید مشخص باشد .
- میکرو Master جهت ارتباط با Slave پایه $\overline{SS}$ را به طور اتوماتیک صفر می کند .
- اگر بخواهیم چند Slave را با یک Master کنترل کنیم پین $\overline{SS}$ در Slaveها را به پین های مختلف Master وصل می کنیم و در صورت تمایل به ارتباط با هر کدام از Slave ها پایه $\overline{SS}$ ان را صفر کرده و بقیه را یک می کنیم .
- با کنترل بیت آخر رجیستر SPSR(SPI status register) می توان از ارسال کامل یک بایت مطلع شد (وقتی که یک بایت داده ارسال شود این بیت یک می شود)

رجیستر SPDR (SPI data register) :

SPDR رجیستر داده SPI می باشد یعنی نتیجه ارسال یا دریافت درون این رجیستر ریخته می شود . نوشتن بر روی این رجیستر شروع انتقال داده را موجب خواهد شد و خواندن ان موجب خواندن داده موجود در بافر دریافت خواهد شد .

SPDR	7	6	5	4	3	2	1	0
بیت مهم	MSB							LSB

رجیستر SPCR (SPI control register) :

رجیستر کنترلی SPI می باشد که بیت های ان به صورت زیر است :

SPCR	7	6	5	4	3	2	1	0
بیت مهم	SPIE	SPE	DORD	MSTR	CPOL	SPHA	SPR1	SPR0

SPIE(SPI interrupt enable) : فعال ساز وقفه SPI می باشد که در صورت ارسال یک بایت اطلاعات ، وقفه اتفاق می افتد .

SPE(SPI enable) : فعال ساز پروتکل SPI می باشد .

DORD(data order) : این بیت اگر یک شود ابتدا بیت LSB ارسال می شود در غیر اینصورت ابتدا بیت MSB ارسال می شود .

MSTR : در صورت یک شدن این بیت ، میکرو به عنوان master انتخاب می شود .

CPOL : با یک شدن این بیت ، در حالت بیکاری پایه SCR یک شده و در غیر اینصورت صفر می باشد .

CPHA : اگر صفر باشد کلاک ارسال در لبه پایین رونده انجام می شود اما اگر یک باشد در لبه بالا رونده کلاک ارسال انجام می شود .

SPR1,0 : جهت تنظیم کلاک SPI می باشند .

این دو بیت نرخ SCK را که Master ایجاد می کند مطابق جدول زیر تعیین می کنند .

SPI2X	SPR1	SPR0	فرکانس SCK
۰	۰	۰	f_{osc} /٤
۰	۰	۱	f_{osc} /١٦
۰	۱	۰	f_{osc} /٦٤
۰	۱	۱	f_{osc} /١٢٨
۱	۰	۰	f_{osc} /٢
۱	۰	۱	f_{osc} /٨
۱	۱	۰	f_{osc} /٣٢
۱	۱	۱	f_{osc} /٦٤

جدول 1-10

کتابخانه <SPI.h> دارای تابع SPI می باشد به طوری که این تابع می تواند همزمان کار ارسال و دریافت را انجام دهد .شکل دستور به صورت (دیتایی که باید خارج شود) spi= (دیتایی که وارد شده) می باشد . توجه داشته باشید که در ارتباط SPI به هنگام ارتباط

اطلاعات هم وارد و هم خارج می شود . پس اگر می خواهید اطلاعاتی را فقط بگیرید باید اطلاعات را نیز بگیرید حال می توانید از آن اطلاعات استفاده نکنید .

وقفه SPI :

در صورتی که وقفه SPI را فعال کنیم با هر بار ارسال یا دریافت یک بایت cpu به تابع وقفه می رود و در انجا می توان SPDR را خواند (دریافت) و یا روی آن نوشت (ارسال) .

نکته : در صورتی که برای ارسال و یا دریافت از دستور SPI استفاده می کنیم باید به صورت 8 بیتی ارسال و یا دریافت کنیم بدین معنی که اگر بخواهیم واژه "ali " را ارسال یا دریافت کنیم باید به صورت کلمه کلمه این کار را انجام داد .

سه روش برای ساخت تابع دلخواه وجود دارد :

1- برنامه را داخل یک نمایه TXT نوشته و پسوند آن را h. می گذاریم و در محل save پروژه کپی می کنیم . در اول برنامه بعد از include به این صورت تعریف می کنیم :

include" نام فایل .h "

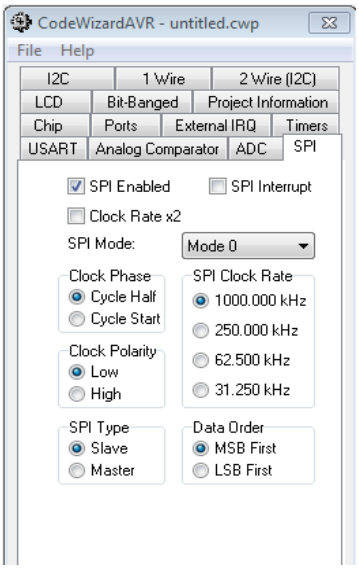
2- برنامه نوشته شده در فایل با پسوند h. در فولدر include ها قرار می دهیم و داخل برنامه مانند بقیه کتابخانه ها معرفی می کنیم :

#include <نام فایل.h>

3- در داخل برنامه تابع دلخواه تعریف می کنیم .

تنظیمات CodeWizard :

در قسمت SPI گزینه ای با نام SPI Enable وجود دارد، آن را فعال کنید .



بعد از فعال کردن آن پنجره به صورت زیر در می آید :

SPI Interrupt : وقفه SPI را فعال می کند .

Clock Rate x2 : سرعت تبادل اطلاعت را دو برابر می کند.

SPI clock Rate : سرعت کلاک SPI را تعیین می کند .

Clock polarity : اگر در حالت high باشد در حالت بیکاری پایه sck یک می شود .

Clock phase : تعیین ارسال اطلاعات در لبه پایین رونده یا بالا رونده. **شکل 2-10**

Data order : مشخص می کند ابتدا بیت LSB ارسال شود یا MSB ارسال شود .

Mode : با انتخاب هر یک از مد ها تنظیمات ان مطابق پیش تنظیمات codewizard تغییر خواهد کرد .

مثال : بکمک پروتکل SPI بین دو میکرو دو کلمه ali zarei و fariborz jowzi را جابجا نمائید.

برنامه Slave //

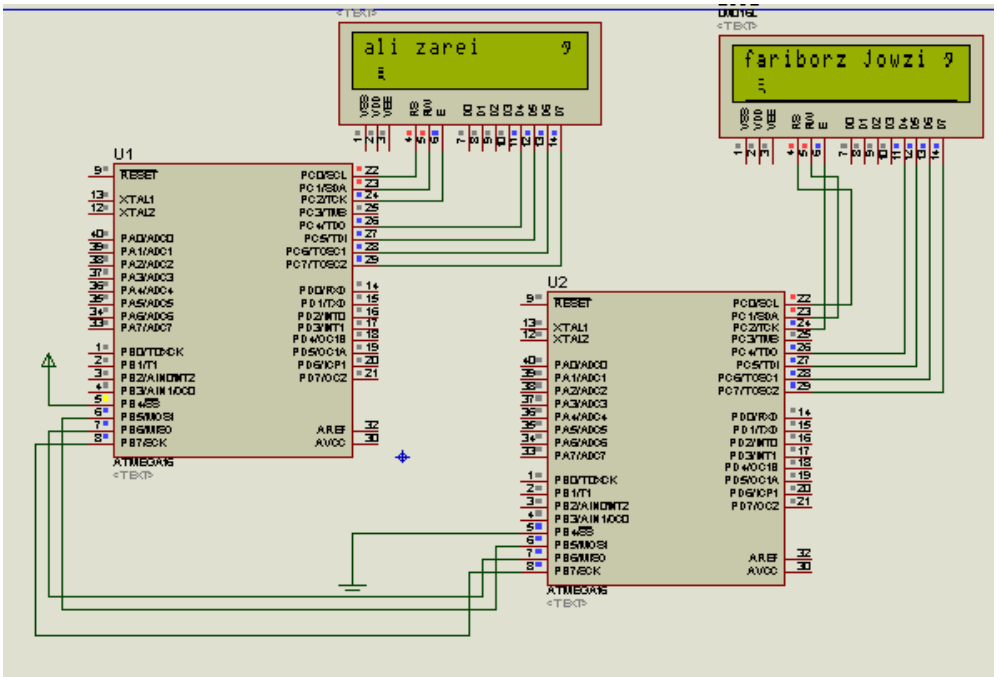
```
#include <mega16.h>
#asm
  .equ __lcd_port=0x15 ;PORTC
#endasm
#include <lcd.h>
```

```
#include <spi.h>
#include <delay.h>
unsigned char a[]="ali zarei",b[14],i;
void main(void)       {
lcd_init(16);
while (1)         {
    for(i=0;i<8;i++)         {
    b[i]=spi(a[i]);
    lcd_gotoxy(0,0);
    lcd_puts(b);
    delay_ms(100);              }}}
```

برنامه Master //

```
#include <mega16.h>
#asm
  .equ __lcd_port=0x15 ;PORTC
#endasm
#include <lcd.h>
#include <delay.h>
#include <spi.h>
```

```
 unsigned char a[]="fariborz jowzi",b[14];
void main(void)          {
lcd_init(16);
while (1)        {
    for(i=0;i<14;i++)          {
     b[i]=spi(a[i]);
     lcd_gotoxy(0,0);
     lcd_puts(b);
     delay_ms(100);       }}}
```



شکل 3-10