

session10 notes

Spi : serial peripheral interface

ارسال داده: full duplex است یعنی هم زمان هم میتواند داده ارسال کند هم دریافت کنه چون دو تا خط جداگانه برای ارسال و دریافت داده داره.

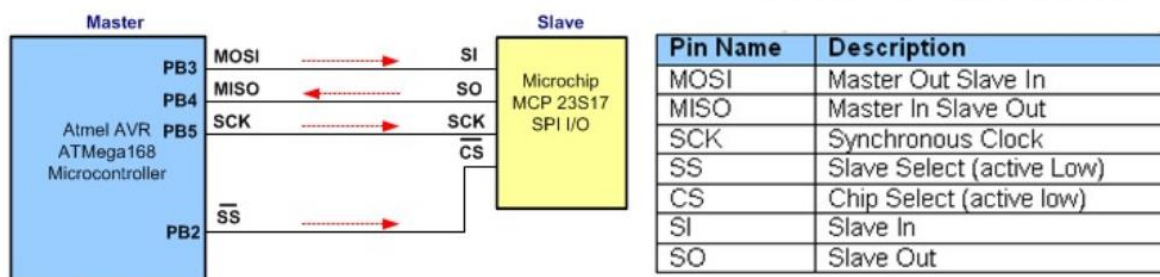
ارسال داده به صورت سنکرون است یعنی ما یک کلاک برای سنکرون کردن اطلاعات داریم. توی ارتباط سریال یا spi ما ۴ تا سیم داریم.

۱. سیم مربوط به کلاک: sck

۲. سیم مربوط به ارسال داده از سمت مستر: MOSI = MASTER OUTPUT SLAVE INPUT

۳. سیم مربوط به دریافت داده از سمت اسلیو: MISO = MASTER INPUT SLAVE OUTPUT

۴. slave select که با SS نشان میدهند.



Typical SPI Master and SPI Slave Device Connection

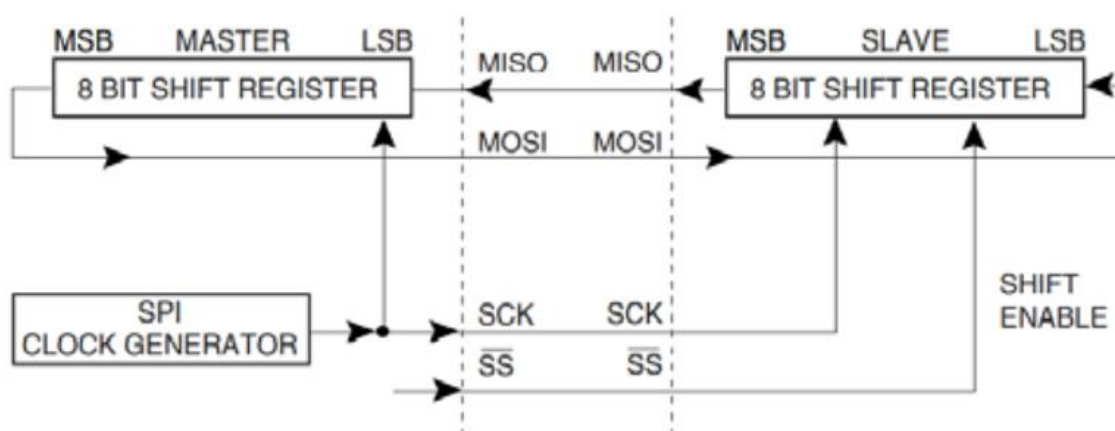
مستر کلاک را برای اسلیو میفرسته و در این بازه، اطلاعاتی که قراره بفرسته را روی خط MOSI قرار میده و اگه هم قراره اسلیو داده هاشو برای مستر بفرسته، روی خط MISO ارسال میکنه.

در یک سیستم spi این مستر هست که کلاک را برای اسلیو میفرسته و اسلیو هیچ کنترلی روی کلاک نداره و سیگنال slave select از طرف مستر ارسال میشه و اسلیو به محض شناسایی این سیگنال، داده هاشو روی کانال برای مستر قرارمیده.

ساختار spi

هم در مستر هم در اسلیو به شیفت رجیستر ۸ بیتی داریم که به خط های miso, mosi متصل هستند.

یه بخش clock generator هم داریم که کلاک بخش spi رو تعیین میکنه.



شکل 9-2: نحوه انتقال اطلاعات در ارتباط SPI

مستر یه داده ای را توی ۸ بیت شیفت رجیسترش قرار میده به محض اینکه داده توش نشست، کلاک هم تولید میشه و به ازای هر کلاکی، یک بیت شیفت داده میشه و برای اسلیو ارسال میشه و داخل شیفت رجیستر قرار میگیره.

به ازای هر کلاک که یه بیت وارد اسلیو میشه یه بیت ازش خارج میشه و میره داخل شیفت رجیستر مستر میشینه

بعد از ۸ کلاک به نظر میاد که کل داده ی مستر با داده ی اسلیو جابه جا شده.

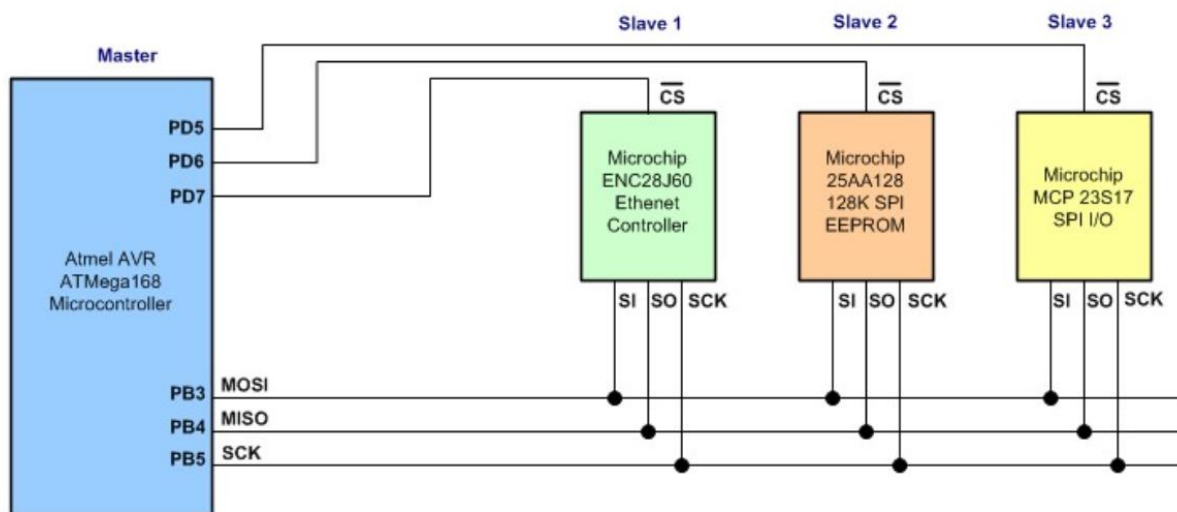
به محض اینکه یه بایت یا ۸ بیت داده وارد مستر شد پروسه انتقال تمام میشه بلافاصله یه فلگی فعال میشه و میتونه وارد زیر برنامه ی اینترایت بشه و یا اینکه اعلام کنه به مستر که این داده آماده هست و میتوانیم برش داریم.

پس توی روتین اینترایت مستر میتوانیم داده ای که توش نشست را برداریم و پردازش کنیم (داده ای است که از طرف اسلیو اومده)

همچنین به محض اینکه ۸ بیت داده مستر توی اسلیو نشست، یه اینترایتی تولید میشه و اسلیو میره داده رو برمیداره.

قبل از شروع پروسه ی بالا:

Slave select باید شروع بشه و یه سیگنالی فعال بشه تا اسلیو متوجه بشه آیا اطلاعاتی که داره از مستر ارسال میشه برای خودش است یا برای اسلیو دیگه ای.



Typical SPI Master with Multiple SPI Slave Device Connection

یه اتمگا ۱۶ به چندین دستگاه خارجی متصل شده و برای هرکدوم از این ماژول ها، خط کلاک و miso,mosi وصل شده.

هرکدام از این ماژول ها chip select ای دارند.

مثلا اگه بخاد داده را بفرسته برای eeprom فقط چیپ سلکت مربوط به اون رو فعال میکنه و اطلاعاتش را از طریق ۳ تا سیم پایینی میفرسته.

تعداد دیوایس های spi که میشه به میکرو متصل کنیم به تعداد پایه های آزاد در میکرو وابسته است که باید اون پایه هارو مثل شکل بالا به عنوان چیپ سلکت استفاده کنیم.

مثلا اگه یه میکرو ۱۰ تا پایه ی آزاد داشت ما میتوانیم ۱۰ تا دستگاه یا ماژول را بهش وصل کنیم.

چون هر دستگاهی عملا ۴ تا خط را به خودش اختصاص میده نصب این سخت افزار سخت میشه.

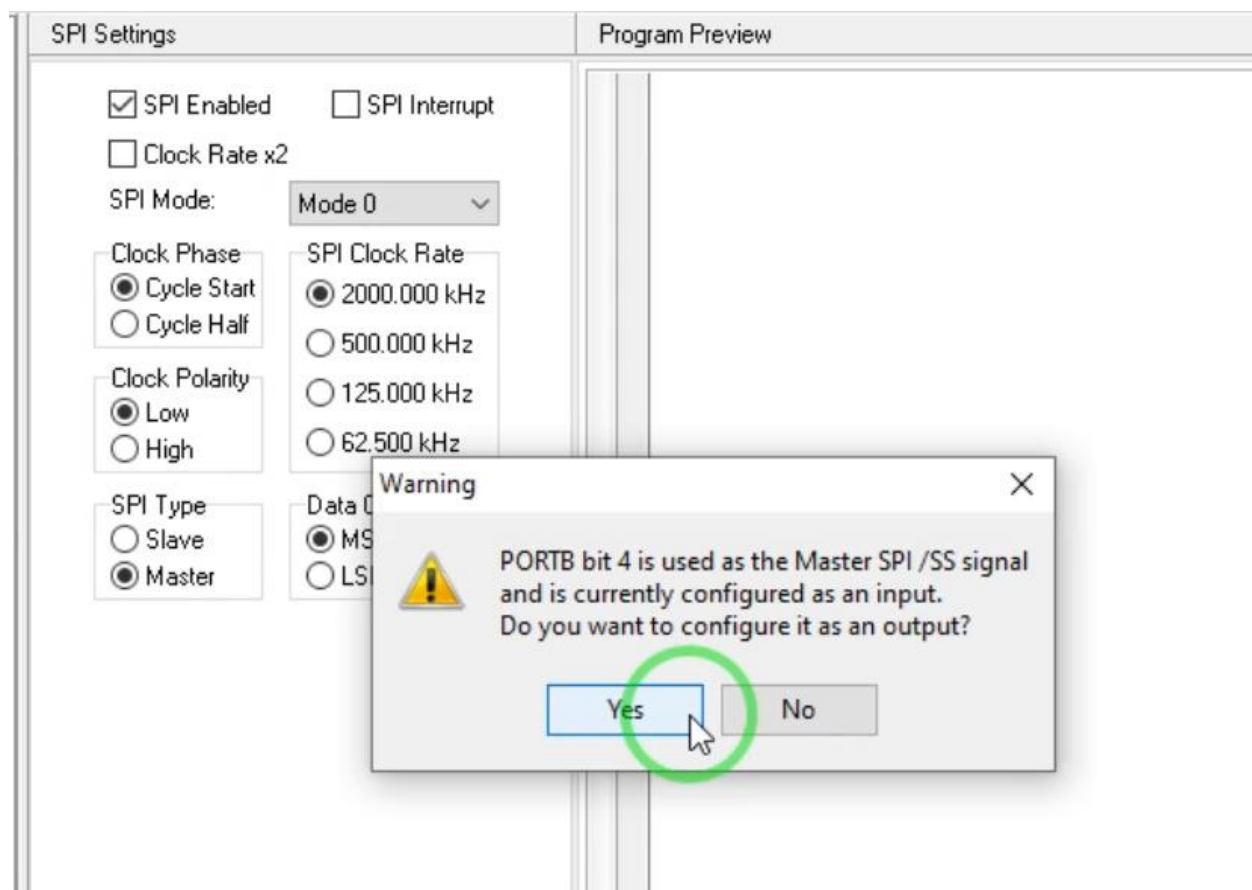
شیفت رجیسترها:

SPDR: داده توی این ذخیره میشه

نکته: اگر ارتباط SPI بین دوتا میکرو است <= تنظیماتی که توی کدویزارد انجام میدید باید دقیقا مثل هم باشه (کلاک فیز و POLARITY و DATA ORDER)

برای مستر:

کلاک و چیپ سلکت و خط ارسال خروجی است.
خط دریافت ورودی است.

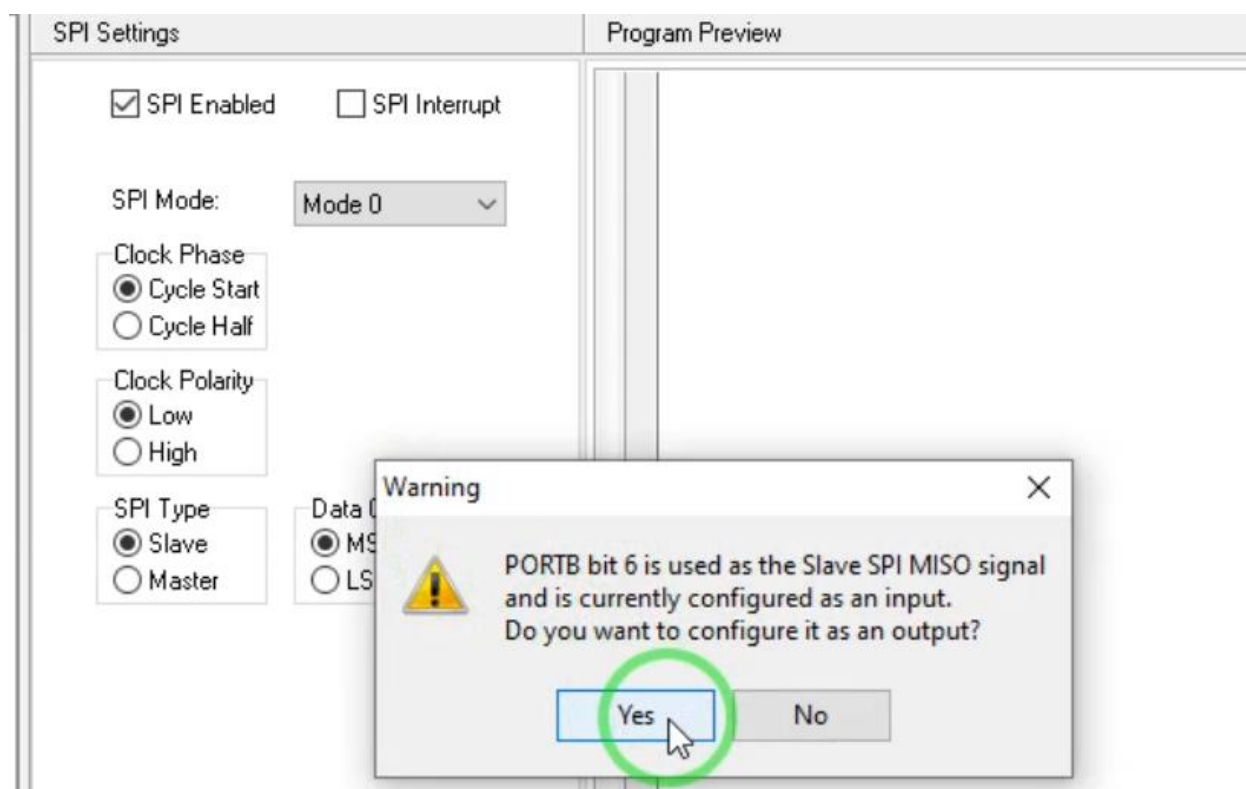


```
spi('z');  
data=spi(0);
```

transfer data
recieve data

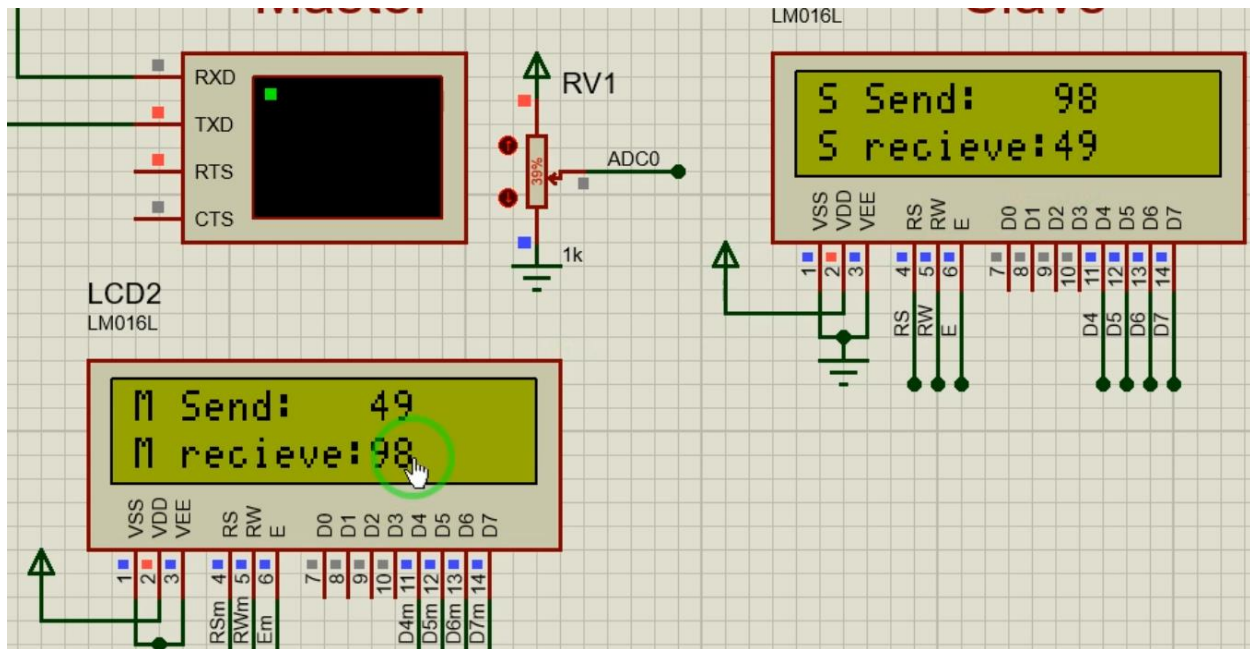
همواره برای ارسال و دریافت داده باید به اطلاعاتی را توی شیفت رجیستر مستر قرار بدیم.

چون به محض نشستن اطلاعات توی شیفت رجیستر کلاک فعال میشه و این کلاک باعث



```
5 // SPI Data Order: MSB First
6 SPCR=(0<<SPIE) | (1<<SPE) |
7 SPSR=(0<<SPI2X);
8
9 datain=spi(dataout);
0
1 while (1)
2 {
3     // Place your code here
4 }
```

تفاوت این پروسه با ماستر:



اسلیو تا زمانی ک کلاکی بهش اعمال نشه نمیتوانه اطلاعاتی را ارسال و یا دریافت کنه
ولی مستر به محض اینکه داده توی رجیسترش نشست، با تعیین چیپ سلکت یک اسلیو
خاص اطلاعات را ارسال میکنه