

# Re: Confused about synchronous communications

---

*Source:* <http://sci.tech--archive.net/Archive/sci.electronics.design/2007-09/msg04071.html>

---

- *From:* "Jon Slaughter" <[Jon\\_Slaughter@xxxxxxxxxxx](mailto:Jon_Slaughter@xxxxxxxxxxx)>
  - *Date:* Fri, 21 Sep 2007 16:46:53 GMT
- 

"Jan Panteltje" <[pNaonStpealmtje@xxxxxxxxxx](mailto:pNaonStpealmtje@xxxxxxxxxx)> wrote in message  
[news:fd081u\\$9gs\\$1@xxxxxxxxxxxxxxxxxxxxxx](mailto:news:fd081u$9gs$1@xxxxxxxxxxxxxxxxxxxxxx)

On a sunny day (Thu, 20 Sep 2007 23:03:38 GMT) it happened "Jon Slaughter"  
<[Jon\\_Slaughter@xxxxxxxxxxx](mailto:Jon_Slaughter@xxxxxxxxxxx)> wrote in  
<[e%ClI.8213\\$JD.4197@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:e%ClI.8213$JD.4197@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)>:

Yes, but I don't think that is anything to difficult? I just have to make  
sure I set the lines correctly and wait atleast the minimum times.

Look at the datasheets is all I can say.

I did but was confused when exactly the data on the data line had to be  
correct. They show the data edge right and the clock edge rise at the  
same  
time. This can't work though and I didn't really notice the clock was 2x  
as  
fast ;/ It make sense and everything but I really wanted to make sure  
before I went any farther with my code(which I essentially have the clock  
at  
the same rate as the data(but I was just testing my code with some leds so  
it wasn't critical).

Jon

I dunno what sort of code you have, asm, C, BASIC, or whatever, but for  
i2c  
you could download C code from Philips IIRC.  
I have open source drivers in C too, they are in [iiclib.c](http://panteltje.com/panteltje/xkrs/xkrs-1.1.tgz)  
<http://panteltje.com/panteltje/xkrs/xkrs-1.1.tgz>  
That program does i2c protocol via 3 par port pins.

Thanks,

Re: Confused about synchronous communications

I'll check it out. Unfortunately I have much more work to do because the driver I was using for port communication(Inout32) with C# was taking 10–40ms to call. So basically just sending data to the port continuously would run only at about 50khz – 100khz. Not all that bad but those I'm going to try and figure out how to get it faster.

For your i2c communication it seems like you don't use any interrupts or polling? If that's the case then I shouldn't have any problem. Although I'm reading up on kernel mode drivers in windows at the moment I really don't want to go that route if I don't have to.

Thanks,  
Jon

.