

Re: Confused about synchronous communications

Source: <http://sci.tech--archive.net/Archive/sci.electronics.design/2007-09/msg04085.html>

- *From:* "Jon Slaughter" <Jon_Slaughter@xxxxxxxxxxx>
 - *Date:* Fri, 21 Sep 2007 14:16:59 -0500
-

"Jan Panteltje" <pNaonStpealmtje@xxxxxxxxxx> wrote in message [news:fd0vdn\\$5ru\\$1@xxxxxxxxxxxxxxxxxxxx](mailto:news:fd0vdn$5ru$1@xxxxxxxxxxxxxxxxxxxx)

On a sunny day (Fri, 21 Sep 2007 16:46:53 GMT) it happened "Jon Slaughter" <Jon_Slaughter@xxxxxxxxxxx> wrote in <[1ASli.6266\\$FO2.4616@xxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:1ASli.6266$FO2.4616@xxxxxxxxxxxxxxxxxxxxxxxxxxxx)>:

For your i2c communication it seems like you don't you any interrupts or polling? If thats 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 too.

Thanks,
Jon

i2c has the advantage that it works on a multitasking system (Linux or mswindows).
If the communication is interrupted by a task switch, the system will just wait a few milliseconds.
'polling' does not apply here, as you can change the clock (scl), and then read the input (sda), so you have full control.
The routines have a delay that you can set (soft delay), in a more modern version
in Linux one could use `usleep()`, a non-clockspeed dependent delay.

What I plan on doing is timing how long it takes to send a bit to the port... actually it takes about 3us in my C++ non-managed code using a kernel mode driver and it about 7us in C#. I'll time a delay routine so I know just how much to delay. The only problem is that task switching will, in general, extend these delays quite a bit. But if I don't have to poll then its no problem.

Re: Confused about synchronous communications

Essentially if all communications is governed by the clock line, the master is the only one that can control it, and arbitrary frequency shifts in won't screw up the communications then its no problem.

I believe all are true but I'm not 100%.

Thanks,
Jon