

Re: Timing in Synch Comm.

Source: <http://sci.tech--archive.net/Archive/sci.electronics.design/2007-09/msg03978.html>

- *From:* Don Bowey <dbowey@xxxxxxxxxxxx>
 - *Date:* Thu, 20 Sep 2007 17:41:31 -0700
-

On 9/20/07 4:04 PM, in article pan.2007.09.20.23.04.33.168563@xxxxxxxxxxxxx, "Rich Grise" <rich@xxxxxxxxxxxx> wrote:

On Thu, 20 Sep 2007 22:30:21 +0000, Jon Slaughter wrote:

"Joerg" <notthisjoergsch@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message

...

I believe the OP wasn't talking about SPI. But with SPI you are right. Many devices, even simple DACs and ADCs have time-out circuits in there and when SPICLK hasn't come for a certain period of time during a transmission they will abort for the affected data set.

Actually I'm trying to do it any synch communications. SPI, I2C, and ICSP will be some of the protocols I'll try and implement. Kinda sucks that there is a time out and since SPI doesn't have an acknowledgement it makes it even worse ;/

I believe the OP (that's you, Jon. ;-)) has some confusion about what "synchronous comm" means. Or either I have.

The way I understood it, "sync" means that the receiver provides its own clock, and it gets sync'ed to the master by some scheme.

IMO the only way to do it is to use the received data transitions to sync a clock at 2f and use the "recovered" clock to sample the data at its midpoint. Even if each terminal is synchronized by a master clock traceable to a stratum 3 or better, data detection should be done using the recovered timing.

Re: Timing in Synch Comm.

What you're talking about, driving something from the parallel port, is simply clocked. If you have control over the data and clock, and know how the destination device behaves when it receives them, then just do it; if your receivers are static, they shouldn't care about latency, as long as the signals get there in the right order.

Good Luck!
Rich