

## Re: RS485 Timeout

---

*Source:* <http://sci.tech-archive.net/Archive/sci.electronics.design/2008-01/msg01352.html>

---

- *From:* "Steve" <[sjburke1@xxxxxxxxxxxx](mailto:sjburke1@xxxxxxxxxxxx)>
  - *Date:* Wed, 9 Jan 2008 16:59:42 -0500
- 

<DAXU@xxxxxxxxxxxx> wrote in message  
[news:b19bc9fd-827a-45f4-9f78-7df3347598d1@xx](mailto:news:b19bc9fd-827a-45f4-9f78-7df3347598d1@xx)

Hi,

I am writing an application for a RS485 network. Basically, a client server program.

So when server sends command out, the client who receive it send some data back.

My question is that how long the timeout should be for the server? How long should the server wait before report a time out to user interface?

Many Thanks

Depends on the application, of course. You didn't say if you have control of both ends or not. RS485 is usually implemented as a "master/slave(s)" relationship. You call it client/server system, but is it just two devices? Multiple uncoordinated clients sending inquiries to a common server on RS485 sounds like a collision-fest. I'm assuming your "server" has the role of master if there are multiple "clients".

As long as everyone is operating, there is no impact no matter how long you make the timeout, of course. But if you are polling a bunch of devices and some are dead/unpowered, you can waste lots of bandwidth this way. I usually assume a few hundred millisecs when all slaves are interrupt driven, baud rates in the range of 9.6 - 38k, and their responses don't take too long to create. I suspect others use a much smaller timeout, and use higher baud rates than I do. But it allows me to power up the network in random order and poll all 32 devices in under 10 sec (assuming they are all non-responsive when first tried). After they all stabilize, the latency for any device is under 1 second, and a single missed message runs the latency for all devices out to about 1.25 sec.

Occasionally, with small micro's, I've had the slaves maintain certain long responses "continuously", in background mode, so that when they get a request, they can respond quickly and the timeout can remain small. Some

## Re: RS485 Timeout

internal routines, like C "sprintf", for instance, can take a very long time in real-time or high speed systems. And many can't be called within ISR's, because they are not re-entrant and/or they just keep interrupts disabled too long.

On the flip side, I'll mention the turn-around delay needs to be considered too. Neither end can reply until the previous sender has had time to turn off its driver and free the line. If the reply starts too soon, the first character in the reply will be garbled at the destination. I usually set this time in the 1-5 msec range. Then I turn off the driver in the ISR (allow one extra interrupt after last character is sent). If you have to rely on some handshake method in your foreground code, or a callback, to turn off the driver, you'll probably find that you need longer times, and that the disable time has a lot of variation to it.

Hope this helps,  
Steve