Description:

Instances of SerialCommDevice class, running on both ends of a serial communications channel implement a fault tolerant full-duplex communications path using sequenced packets of printable ascii characters. Using only printable characters avoids problems that can occure with some types of hardware that would otherwise interpret the control characters (such as a modem or a related device).

SerialCommDevice implements two additional useful features. First, a test method provides for remote loopback that can serve to verify the integrity of the interface w/o transfering and data to the remote. Secondly, support for an Out Of Line message allows for temporarily interrupting regular data packet transmission accross the line to supply an asyncronous message to the delegate of the receiving SerialCommDevice instance. SerialCommDevice dedicates a seperate thread that continuously reads the TTY port and the entire object is thread-proof.

serialCommDeviceTest continuously transmits random data and is intended only to verify the data transfer robustness of the SerialCommDevice Class. Just type make, then start the serialRead.csh and serialWrite.csh shells to transmit random data accross the cable. Try disconnecting the cable to see the retrys.

Areas for improvement:

Perhaps the biggest bang for the buck, would be a more efficient encoding mechanism for handling non-printable characters. Another option would be to have a mode where very limited encoding would be performed if the serial channel permited. My inital application for this class transfered only a very limited number of non-printable ascii characters, so the encoding penalty wasn't a concern.

Make the receive side of the interface capable of automatically adjusting to the proper baud-rate. This code is in place & gets invoked if the receiver's init method is provided 'brNone' as the baud-rate. Still needs to be debugged.

How about support for standard encoding mechanism(s).

Have the receiver thread time-out every so often when the line is idle and send (in another short-lived thread) a test packet to ping the remote to verify connection integrity. Along with this, add another delegation method to automatically notify the delegate when the ping fails (line goes down).

kjt

PS: I did all the work on this class at home in my spare time as a quicky play-thing. It just kind-of got out of control...

Contact information:

Ken Taylor
Ken_Taylor@next.com