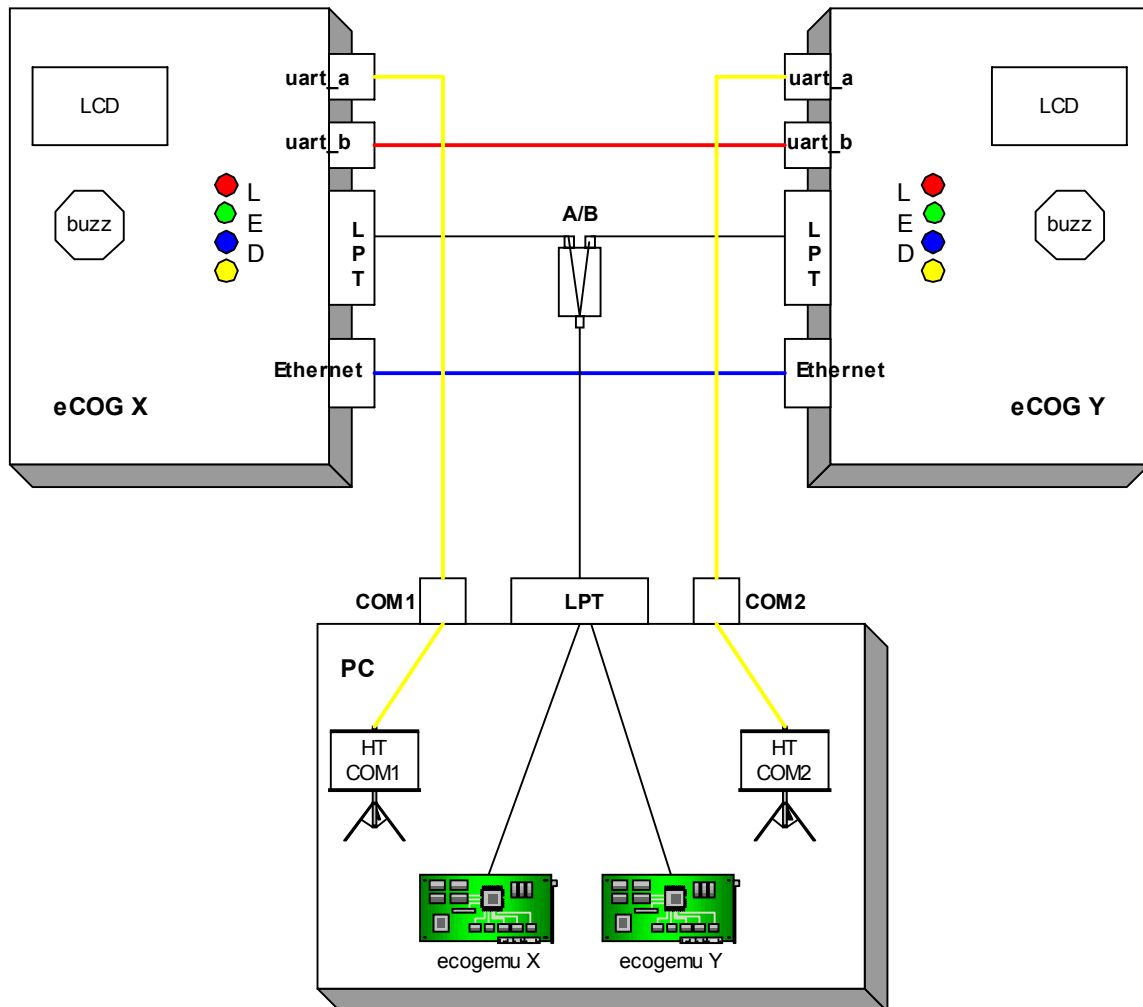


Notes, diagrams, and pictures on labs and PicOS development

Our experiments were conducted on a small farm of eCOG1 eval. boards on an Ethernet LAN, with a Linux server and a PC running Windows and Cygwin. RemoteDisplay in Apps and sfile in Linux directories use this set-up.

All other applets are designed to show PicOS paradigm, with functionality not necessarily related to connectivity. However, because networking is at heart of PicOS design (with prominent TCV) and all our endeavors, most applets can be interconnected. We used the following set-ups:

Simulation / Testing: 2 boards, ecogemu



UART A ports connect to COM ports, and meet HyperTerminals as user interface, UART B and Ethernet ports connect the boards.

Real stuff:



[The boards are running an applet with processes communicating over UART B and Ethernet.]

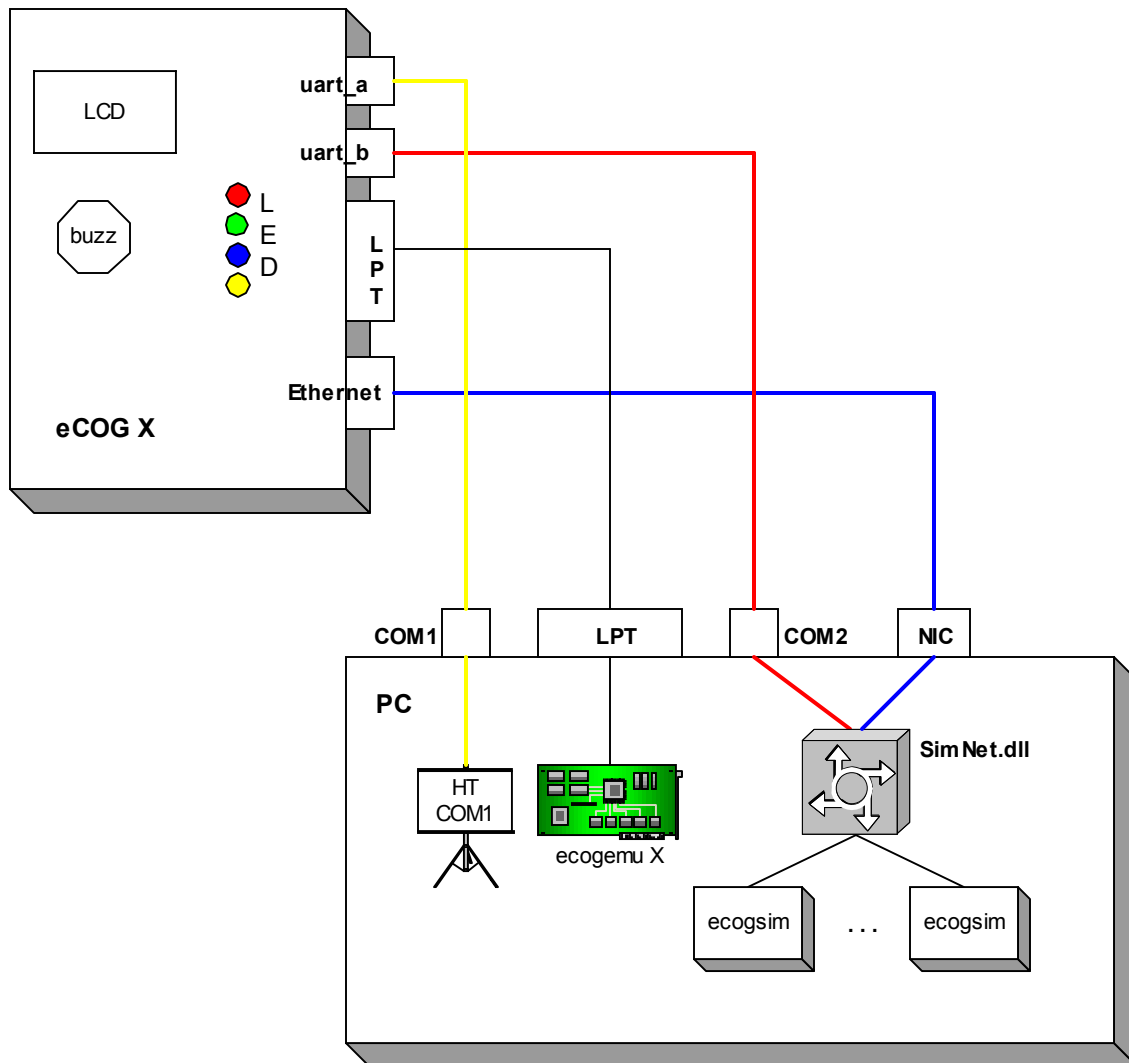
SimNet is designed to prototype, debug, and test applications running on PicOS. Nothing can reveal more details on its device emulation than “*An Emulated IO Subsystem for Reactive Applications*”, available from our *Documents & Downloads* section.

However, no more than setting ECOG_SIM 1 in options.sys is needed to run it for most applications under ecogsim (`cp ../common/app.cmd .; ecogsim app`). Custom command ui “<command>” is equivalent to <command> given over UART A to the same application executing on the eval. board.

When running radio drivers, nodes’ connectivity can be configured under *ecogsim* with custom command *node <i><x><y><r>*, but consecutive *ecogsim* sessions should be “connected” by default, i.e. represent a chain of nodes with RF links between them.

Also, an appropriate dll must be set in the SimNets/Debug directory (e.g. cp SimNet_xemics.dll SimNet.dll; cp SimNet_xemics.exp SimNet.exp; cp SimNet_xemics.lib SimNet.lib), or built with RADIO_TYPE set directly in PicOS/eCOG/sysio.h.

**Simulation / Testing:
1 board, ecogemu, ecogsim**

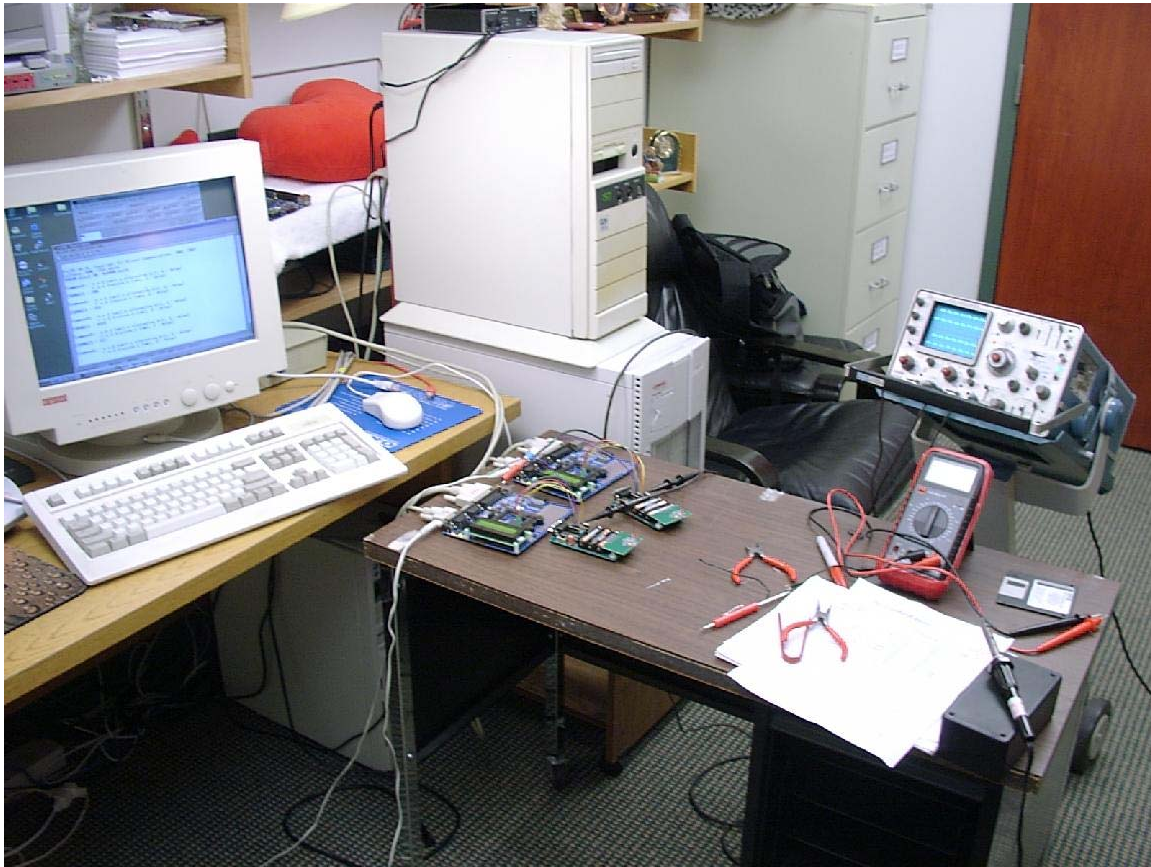


Release 1.0 does not allow for a hybrid set-up – there is no links from SimNet to COM2 or NIC. However, we found the SimNet extremely useful, and if there is a future release on the Cyan's hardware, enhanced SimNet will be part of it. Also, simple extensions to SimNet's user interface help in handling many simulated hosts, as ecogsim's window becomes unbearable. Again, since this release is meant to introduce PicOS, we don't want to blur its prominent features with less material details.

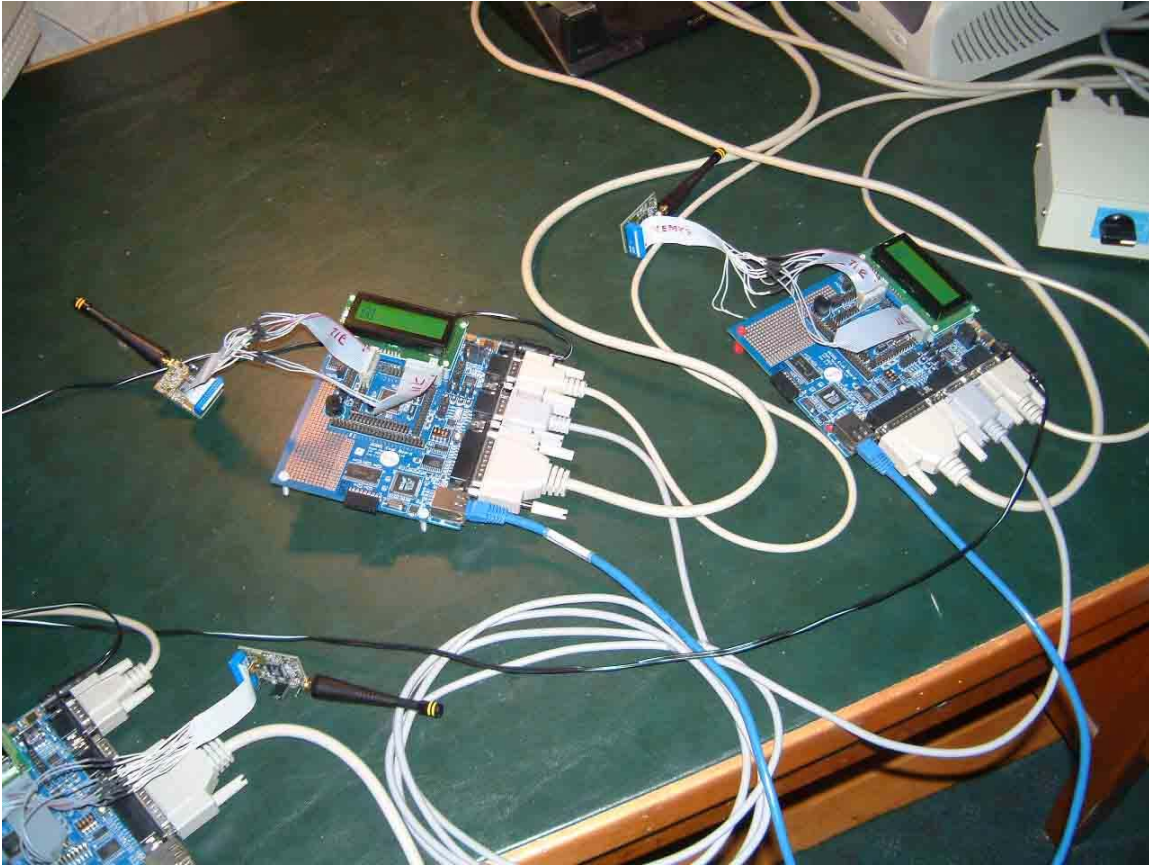
Radio connectivity deserves special attention. We tried three different radio modules, to capture both interrupt-driven and polling principles. We used a proprietary design (from Valert), old modules from RFMI Monolithics (DR-1200DK with TR1100), and several XM1202's from Xemics. (http://www.xemics.com/docs/xe1200/xm1202_prodbrief.pdf)

Pin mappings for each module are listed in *radio.h*. `RADIO_TYPE` in *options.sys* must be set to an appropriate value, one of `{RADIO_XEMICS, RADIO_RFMI, RADIO_VALERT}`, and matching Makefile must be used.

In the labs:



[Smaller boards are DR-1200DK, showing off on the oscilloscope and monitor.]



[Running RFPing over XM1202s]

We tested under Windows '95, '98, 2000, and XP. With Cygwin, any application can be build with Makefile in Apps/common. Without, `nmake -f ../common/app.mak` will do. In case there is no `nmake` installed, we included self-extracting `nmake15.exe` from <ftp.microsoft.com>.

SimNet was built under VC++ 6.0, and all the .dll files are provided in the Debug subdirectory.

Please direct feedback and questions to picos@olsonet.com.