Communication between standalone application and Nios2 or uC/OS-II x86 Port

The Nios II IDE (Eclipse) is using nios2-terminal.exe for communcating with the board.

Text written in the IDE console window is read by nios2-terminal.exe and sent to the board. Nios2-terminal.exe also reads data from the board and outputs it so Eclipse can display it in the console window.

Communicating with the board

We can do the same thing as Eclipse is doing if we are interested in having our own application running on the PC and communicating with the board.

What we have to do is make our own application start a new process running nios2-terminal.exe. Nios2-terminal.exe is reading data from its STDIN and outputting data to its STDOUT so we should also set up pipes for reading and writing to them.

Java sample code:

```
Process p;
BufferedReader in;
BufferedWriter out;
BufferedReader err;
try {
  p = Runtime.getRuntime().exec("N:\altera\kits\nios2\bin\nios2-terminal.exe");
  in = new BufferedReader(new InputStreamReader(p.getInputStream()));
  out = new BufferedWriter(new OutputStreamWriter(p.getOutputStream()));
  err = new BufferedReader(new InputStreamReader(p.getErrorStream()));
} catch(IOException ioe) {
  ioe.printStackTrace();
}
```

It should then be possible to read and write data to/from the board using for example `out.write()` and `in.readLine()`.

Communicating with the port

The same setup can also be used for communicating with the x86 port by replacing nios2-terminal.exe in the code above with the path to the port executable.

For the Windows port I have only succeeded in sending data from the port to the application but not reading data. It seems that it is not accepting data on STDIN or is just reading keystrokes? For the Linux port it works both ways.

Markus Kilås, markus@markuspage.com