

Creating and Running Circuits with the TMU Ports Package

Dave Galloway

*Edward S. Rogers Sr. Department of Electrical and Computer Engineering
University of Toronto*

April 2013

Communicating with your Circuit

The TMU ports package can be used to communicate with your circuit. See *The TMU Ports Package* document for details. If you want to use the ports package, create a port description file, giving it the same name as the top level netlist, but with a .ports extension. You must also instantiate a component in your netlist called `usb_portmux`, passing it the signals that you want to connect to the ports package.

Run the `tmu` command like this:

```
tmu toplevelname
```

This will run the Altera CAD software, creating a new directory called `usb_portmux` containing the Verilog source for the `usb_portmux` wrapper component. It will also copy your ports file to a file called `fpga0.ports`.

Compiling your Circuit

Add this line to your .qsf file, so that Altera's Quartus software can find the source for the `usb_portmux` component:

```
set_global_assignment -name SEARCH_PATH usb_portmux
```

Compile your circuit using Quartus.

Using the demonstration board

In order to use the demonstration board, the `tmumon` daemon must be running on the machine named `skynet.eecg`. This daemon is normally started when the machine is booted, and will continue running. If the daemon stops for some reason, you can re-start it by signing on to that machine and typing:

```
tmumon &
```

The `tm` command can then be used to communicate with the `tmumon` daemon. See *The tmumon Program* document for details.

You can check the status of the demonstration board remotely by running the `tmstatus` program. It will use X-windows to create an popup window on your screen that shows the current status of the board.

The `tmget` command will attempt to reserve the board for a specified number of minutes (default 10). If it succeeds, your name will be shown on everyone's `tmstatus` display. When you are finished, run the `tmrelease` command so that others may use the board.

Once you have ensured that no one else is using the board, you can download your circuit to the board using Quartus. To communicate with the circuit, run your program that uses the ports library in the directory that contains the `fpga0.ports` file.