DE0 Controller GUI

# What you will need:

* Python 3.4 (https://www.python.org/)
* TerasIC DE0 Development Board
* Provided glue logic or compatible glue logic (please see “Glue Logic Guide”)

# Setting up the Program

## Windows

The first thing that needs to be done, is to change the Start\_Server.lnk properties.

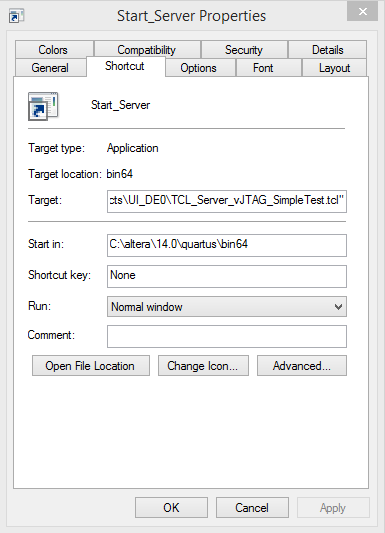
1. Under the Shortcut Tab, change the Target to point to the quartus\_stp.exe and the provided TCL\_Server.tcl in the following manner:

Target:

C:\altera\14.0\quartus\bin64\quartus\_stp.exe -t C:\<Path to TCL\_Server.tcl>"

Start in:

C:\altera\14.0\quartus\bin64

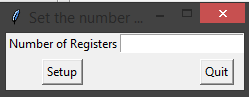


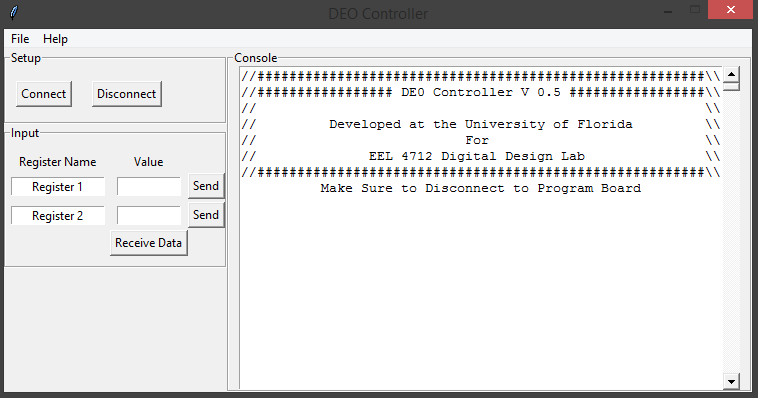
Some things to keep in mind about this:

* The version of Quartus should not matter. It has been tested with version 9, 13 and 14. With any version later than 13, you cannot program the Cyclon III FPGA, as it is no longer supported.
* If you move the TCL\_Server.tcl file, you must update the shortcut file.

# Using the GUI

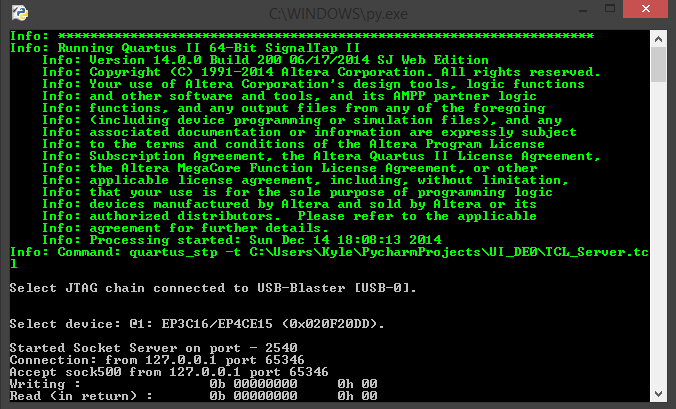
1. To run the program, simply open the main.py file. This will launch the GUI and the Python terminal.
2. When the program first opens, it will ask you how many registers are to be addressed.
   1. All addressed registers will start at 0x01
   2. The last address will always be the return address (0xFF)





1. Once the number of addressed registers are set and your DE0 board is already programed, you will then click on the connect button. This will run the TCL\_Server and open the quartus\_stp.exe. This will take several moments and will then run 100 reads and writes.

**NOTE**: If you need to reprogram your board, you must disconnect your board.



1. Once step 3 is complete, you are ready to start sending and receiving data to the DE0 Board.
2. To send data to the board, simply select the register you want to send data to, enter in the value and then click send.
   1. You will see the value sent to the register appear in the right side of the GUI, and will also see the value appear in the terminal.
      1. The terminal will show 2 read and write sequences. When you are sending data, it will show the address in the first sequence and then the data in the second. When reading back the data, you will first see the address being sent and the second sequence will be the return value.

|  |  |
| --- | --- |
|  | Write to address 1 of value 2  Write to address 2 of value 3  Read the value out of address FF with the add of address 1 and 2 (5) |

# How it works

The GUI is written in python and runs a TCL, TCP/IP server to communicate with the board. The port that is opened is 2540. Once the connection is made to the board from the computer, the python scripts will send binary data to the port number listed. At this point, the TCL script takes over and sends the data in serial. With the TCL script, there are a few main commands that are used that are supported by Quartus.

|  |  |
| --- | --- |
| Open\_device –hardware\_name $usbblaster\_name –device\_name $test\_device | This command opens the JTAG Device so that it will accept further commands |
| device\_virtual\_ir\_shift –instance\_index 0 –ir\_value 1 –no\_captured\_ir\_value | Setups up to send in data |
| device\_virtual\_dr\_shift –dr\_value $send\_data –instance\_index 0 -length 8 –no\_captured\_dr\_value | Data to be sent |
| device\_virtual\_ir\_shift –instance\_index 0 –ir\_value 0 –no\_captured\_ir\_value | Set IR register back to 0 which is bypass mode |
| device\_virtual\_dr\_shift –dr\_value $send\_data –instance\_index 0 -length 8] | This command sends data into the device and also reads back in a value from the tdo pin. |

For a further breakdown, please refer to the document “vJTAG Breakdown”, as well as “vJTAG Megafunction.” Also, for more on the TCL instruction, please refer to the document title “IR Instructions.” You can also find further information on the vjtag, TCL scripting for Altera, and a breakdown of the DE0 Board schematics in the “Supporting Documents” folder.