Georgia Institute of Technology

Tutorial on Embedded Systems - Module III: Writing Your Own C Code

Greg Drayer
VIP Program

Outline

- This module presents the following content:
  - Getting started – materials and software
  - Setting up VNC access
  - Writing your C code
  - Compiling your C code
  - Installing the PCIe driver
  - Running your C application
  - Uninstalling the PCIe driver
  - Safe shutdown
  - Summary

Objectives

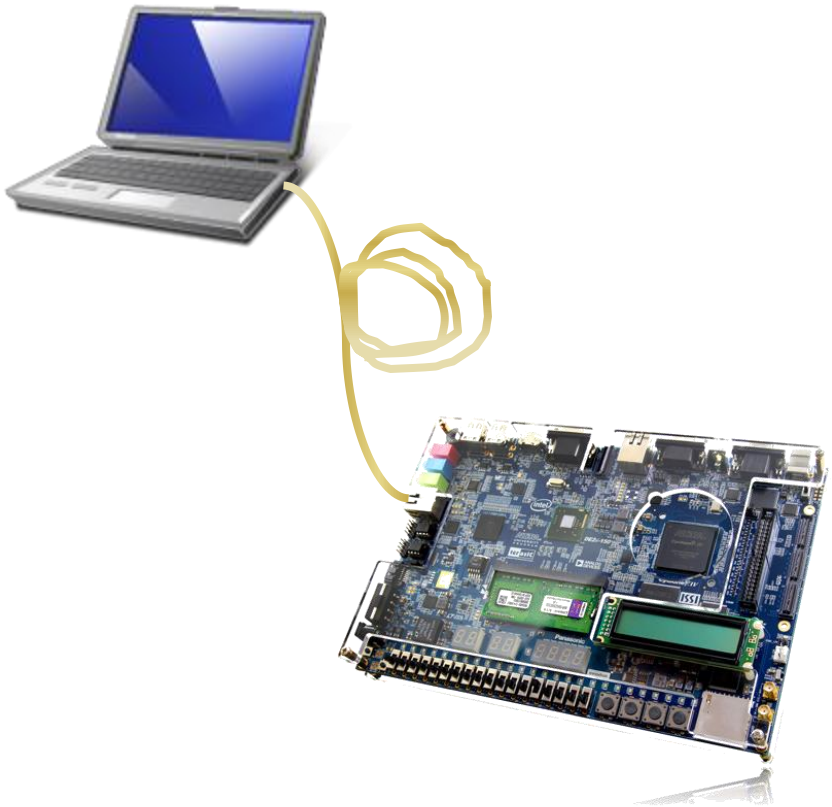
- By the end of this module, you will:
  - have written your own C code for an application that enables interaction between buttons and LED's making use of
    - the Cyclone FPGA
    - the Atom processor
    - PCIe high-speed communication
  - have analyzed the behavior resulting from your code and identified the main functions

Getting Started

- List of materials and software:
  - Laptop or desktop running Windows (XP at least)
  - DE2i-150 development board
  - Power adapter and cord
  - Ethernet cable

Setting up VNC Access

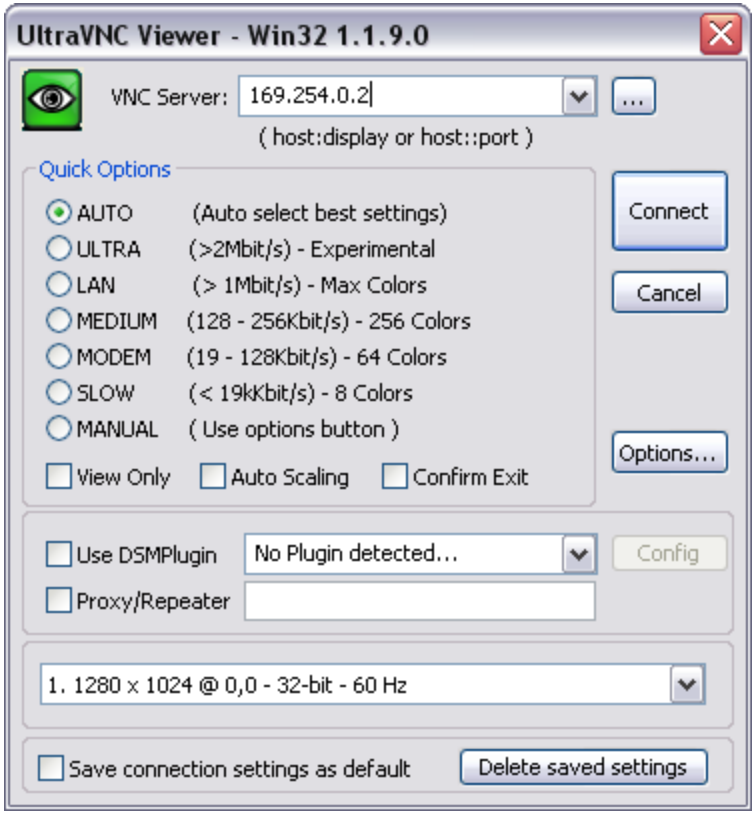
- Connect the ethernet cable between your laptop and the board



- Power up the board
  - Your laptop ethernet network should display “Limited or no connectivity”

Setting up VNC Access

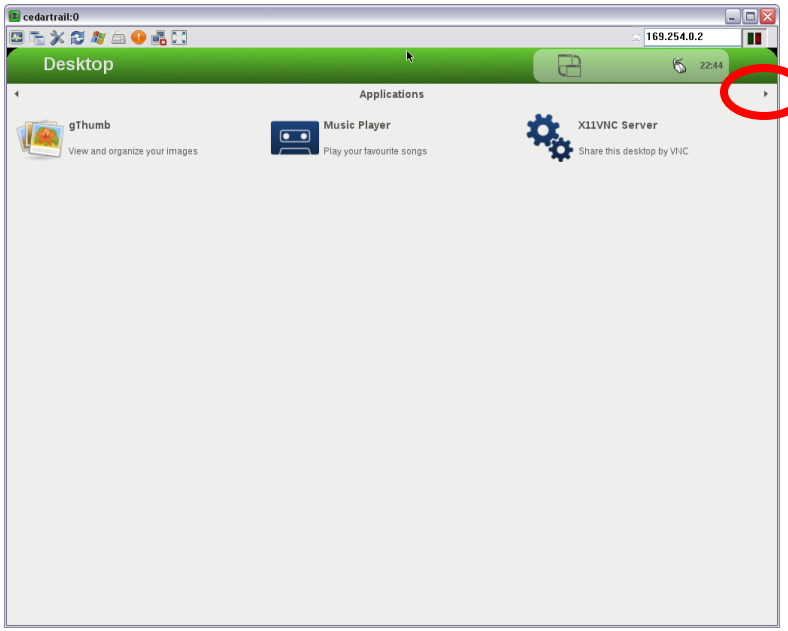
- Connect using UltraVNC
  - Start the UltraVNC on your laptop
  - Set the target VNC server to 169.254.0.2
  - Click on the “Connect” button
  - A window should open showing the Yocto desktop environment

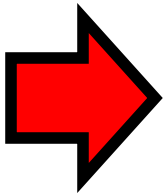


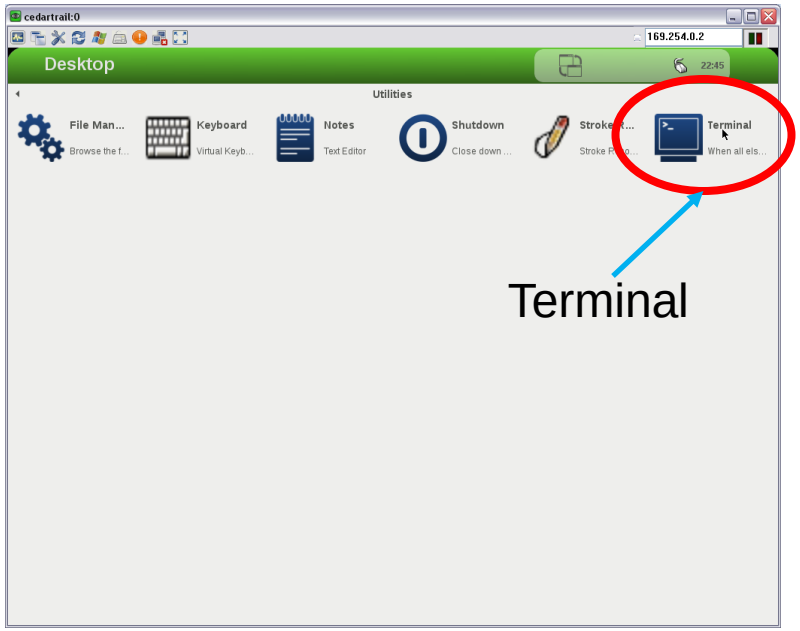
Georgia Institute of Technology

Writing your C Code

Click for the next screen







Writing your C Code

- Go to /home/root/Projects/
  - Type the following on the command line of the Terminal window
    - `cd /home/root/Projects/`
- Create a new folder called “VIPmodule3”
  - Type the following on the command line
    - `mkdir VIPmodule3`
- Create a new file called “app.c”
  - Type the following
    - `leafpad app.c`
- Write the code provided in a piece of paper
  - Make sure to save periodically with Ctrl-S
  - When finished, quit the Leafpad editor with Ctrl-Q
  - From the code written, what behaviors do you expect?

Compiling your C Code

- Place the PCIe libraries in your project's directory
  - Bring the following files from the previous module's folder
    - PCIE.h
    - PCIE.o
    - teraisc_pcie_qsys.so
    - TERASIC_PCIE.h
  - You can do this from the terminal or with the file manager

- Compile your C Code
  - Type the following commands in the terminal
    - `gcc -g -Wall -c app.c -o app.o`
    - `gcc -g -Wall app.o PCIE.o -o app -ldl`

Installing the PCIe Driver

- Go to /home/root/BoardSetup/linux/PCIe_DriverInstall
  - Type one of these two options
    - `cd /home/root/BoardSetup/linux/PCIe_DriverInstall`
    - `cd /home/root/Downloads/BoardSetup/linux/PCIe_DriverInstall`

- Load the PCIe driver
  - Type
    - `sh ./load_terasic_qsys_pcie_driver.sh`
    - The message “Matching Device Found” should appear
    - If it does not appear, proceed to safely reboot the board

- Verify the driver is loaded
  - Type
    - `lsmod`
    - You should see a list of the kernel modules loaded, including the “terasic_qsys_pcie” driver

Running your C Application

- Go back to the application folder
  - Type
    - `cd /home/root/Projects/VIPmodule3`
- Execute the application compiled
  - Type
    - `./app`
- Describe the behavior of the application and the DE2i-150 board
  - Does it comply with your expectations from looking at the code earlier?
- Exit the application
  - Select the option `99` from the menu

Uninstalling the PCIe Driver

- Unload the terasic_qsys_pcie driver
  - Type
    - `rmmod terasic_qsys_pcie`
- Verify that the driver has been unloaded
  - Type
    - `lsmod`
    - You should not see the terasic_qsys_pcie driver on the list

Safe Shutdown

1) Type exit on the Terminal window

2) Close the UltraVNC window

3) Press and hold the power button until the board shuts down completely

4) Store the parts in the bags and boxes for the next class

Summary

- In this session you have:
  - accessed the Yocto desktop environment on the board from your laptop as a VNC client
  - written, compiled and run a custom C program on the DE2i-150 Development Board
  - loaded a PCIe driver for high-speed communication between the Intel Atom processor and the Altera Cyclone IV FPGA
  - interacted with the inputs (buttons) and outputs (LED's) of the board from the Yocto enviroment through a pre-loaded FPGA bitstream