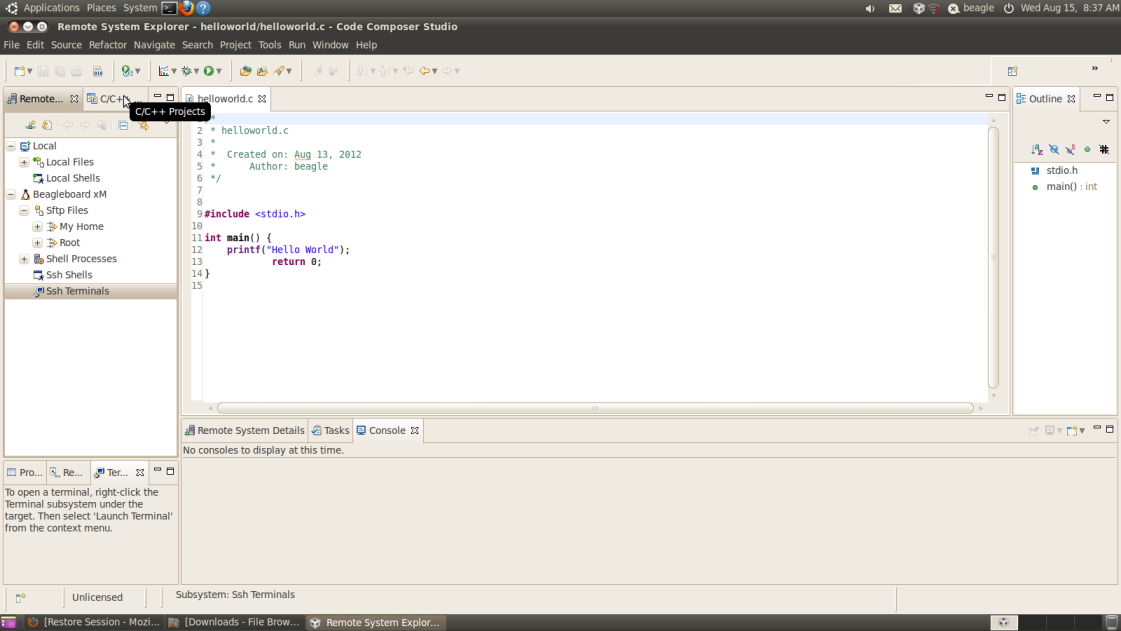
1. 2.1 Open DEV enviroment
   1. Open Vmware Player
   2. Select the Beagleboard Dev image
      * Select edit virtual machine settings
        1. Confirm there is a serial device listed in hardware
           1. If no device listed

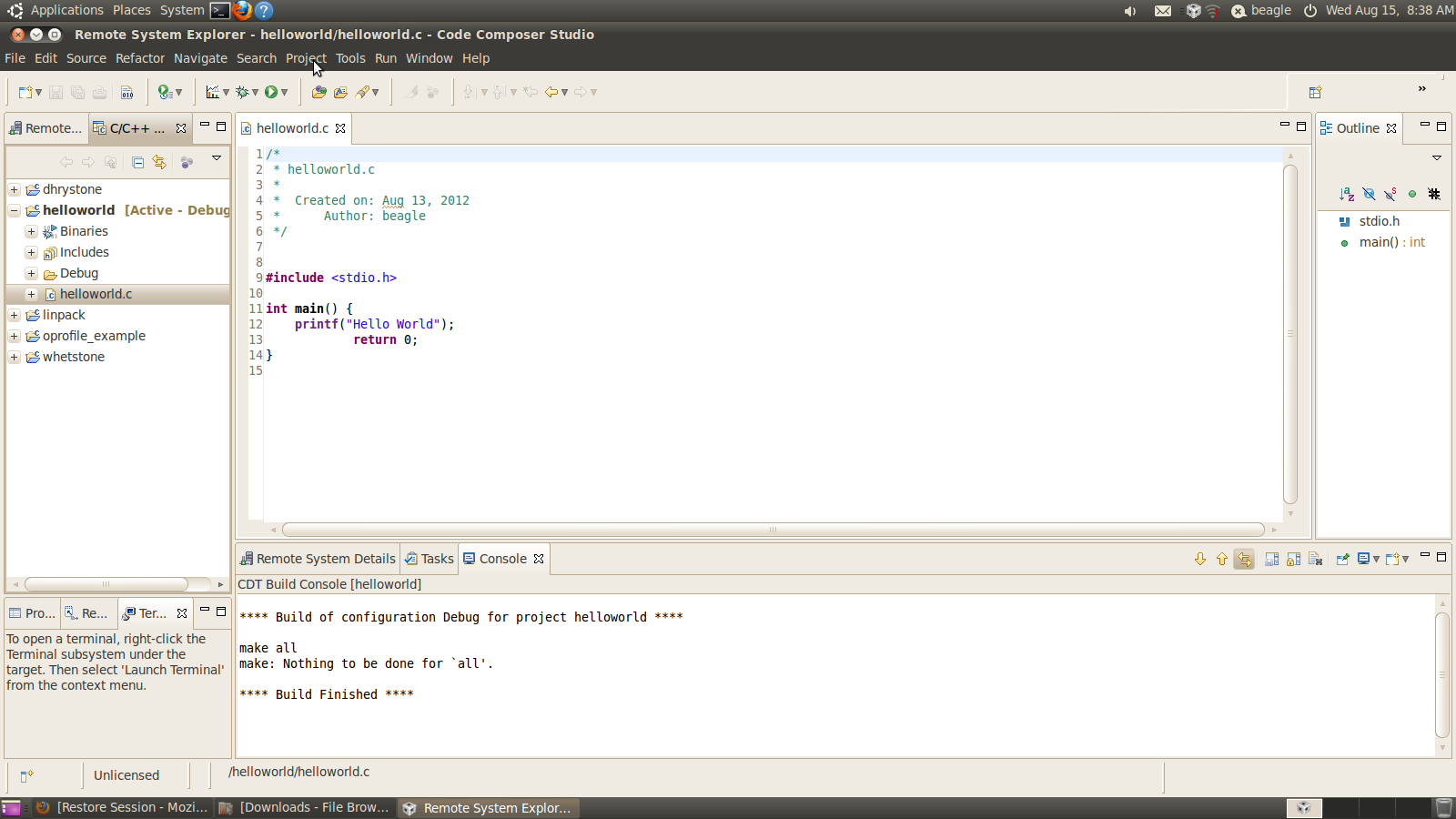
Click the add button on the hardware tab

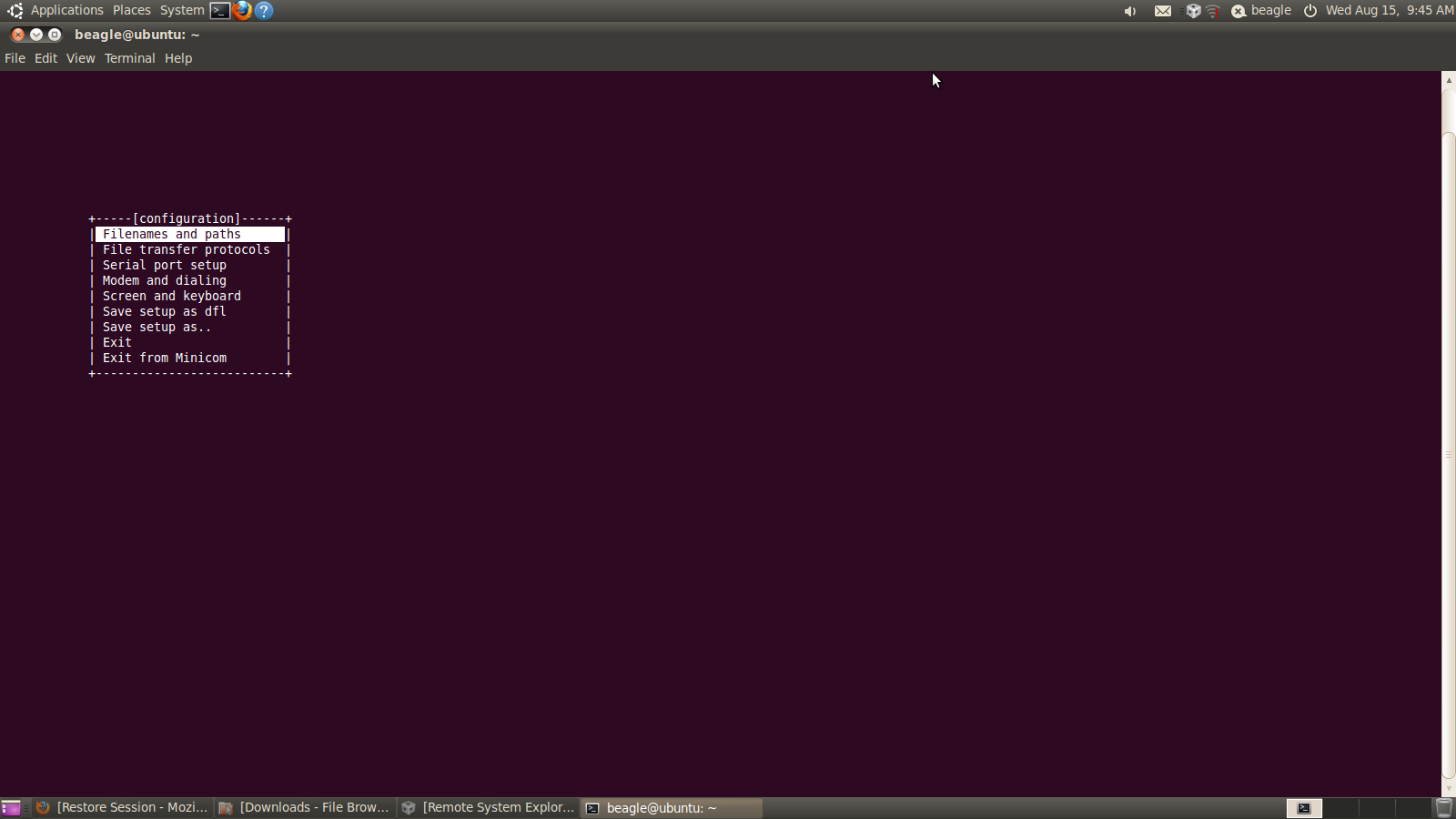
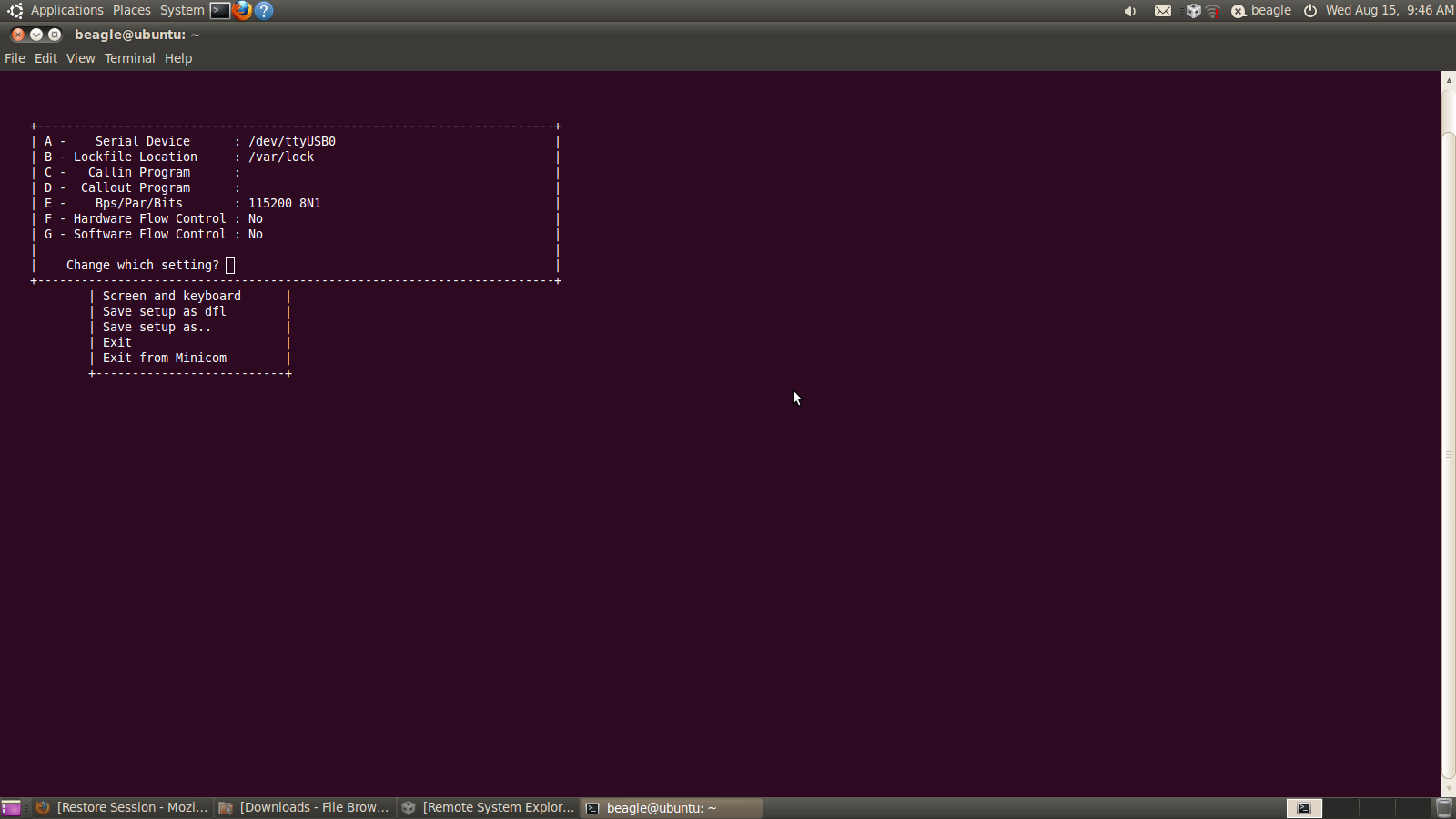
Select serial device

Select Use Physical device and hit next and finish

* + - 1. Confirm the Network adapter is bridged and generate a new Mac address
         1. To generate a new mac address click on network adapter
         2. Then select advanced then click on generate next to the address
  1. Select the Code Creator 5 software
  2. Select C/C++ Projects



* 1. Select the Hello world project and then in the project menu select project build to ensure there are no errors

1. 2.3Hook up Beagleboard
   1. Remove BeagleBoard from packaging
   2. Attach HDMI cord to beagleboard and attach cord to display(Note do not attach if beaglboard is currently powered!!!!)
   3. Ensure an SD card is installed
   4. Attach ethernet, Mouse and keyboard(Headphones if you choose )
   5. Plug in power to boot board
2. 2.4 Minicom
   1. Open up a terminal and type sudo minicom -s to open the minicom setup script
   2. select serial port setup 
   3. Verify that minicom is setup as in the picture with one exception. The correct serial device is /dev/ttyS0
   4. this sets up serial terminal access to the beagleboard which will allow us to diagnose any issues during boot up before linux finishes loading. Press the reset button on the beagleboard to begin booting and you will see the kernel loading and the
   5. log into the board by typing root
      * type in **/etc/init.d/matrix-gui-2.0 stop**  this will turn off the gui browser being displayed on the HDMI output
      * From this root prompt we can run programs on the board communicating over the serial prompt but that will not be our preferred method.
3. Tftp BOOT AND NFS
   1. open a terminal Type **ifconfig** in you development enviroment
   2. This should give you an IP address of 192.168.1.XXX
   3. If you do not get a ip address ensure your host system is connected to the cisco router and
   4. Then run **sudo dhclient -r** and **sudo dhclient** to renew your ip address
   5. cd to the ti-sdk\* folder in your home directory
   6. type in **sudo ./setup.h** this will run the ti setup script
      * You will be asked which directory do you want to use as the nfs share select the default directory the next question it will ask is if you want to overwrite the existing files select skip. Do the same thing for tftpboot
      * It will ask for the correct serial port it should default to the correct one if using the serial port on the Desktop
      * It will prompt for your ip address make sure the ip is correct
      * It will then ask how you want to boot select tftp and nfs
      * You will then be asked if you want to save the script select yes, you are then asked if you want to run it select No
   7. There is a template uEnv.txt document in the home folder
      * Edit this txt file so that the server IP is the ip listed in your virtual machine
      * Ensure that the ipaddr that you use is of the form 192.168.1.21X where X is your group number and the gateway address should be 192.168.1.1
      * Look at the setup script you created and ensure that the file name listed for tftp boot and the directory listed for nfsboot are copied into uEnv.txt
      * Power off the beagle board and remove the sd card from the beagle board and plug the sdcard adapter into the development machine and rename the existing uEnv.txt as uEnv\_SD and paste the new tftp version into the fat partition with the newly edited version.
      * Watch the boot up process and confirm that you are booting from the nfs system using **cat /proc/cmdline**
4. 2.5 RSE explorer.
   1. Open the Code Creator and select the remote system explorer option
   2. Right click on the Beagleboard Xm device listed and change the hostname to the ip that you entered in uEnv.txt
   3. Open the C/C++ projects folder , copy the helloworld binary and paste the hello world binary into /home/root (Note this can also be accomplished pasting directly to the target nfs directory)
   4. Then go back to the remote system explorer
      * Launch a Terminal by right clicking terminals and selecting Launch terminal
      * Change to /home/root directory
      * Chmod +x helloworld
      * ./helloworld
5. 2.7 Breakpoint
   1. Open up a terminal in the Remote System explorer and start a gdb server on the target board
      * Command : gdbserver :10000 helloworld Debug
   2. Go back to C/C++ projects and right click the helloworld binary and select debug as
   3. Select C/C++ remote application
   4. Name the config HelloWorld\_Debug
   5. Change the Connection to Beagleboard xm
   6. Change the remote directory to /home/root
   7. In main hit the select other option
      * Choose manual Debugging and hit ok
   8. Change to the debugger tab
      * Change the debugger to /home/beagle/ti-sdk-beagleboard-05.05.00.00/linux-devkit/bin/arm-arago-linux-gnueabi-gdb
      * Change gdb command file to /home/beagle/ti-sdk-beagleboard-05.05.00.00/.gdbinit
      * Select the connection tab in the debugger tab
        1. Make sure the correct IP Address is listed
        2. Make sure port 10000 is listed as the port
      * Hit apply and then debug
   9. You should be halted at the main function
   10. Press the step over button to print hello world
       * If you do not see this in your console goto window->Navigation-> previous view to access the remote system explorer and you will see it printed there
   11. Hit the run button to finish execution and return to the RSE view
   12. Add a couple extra print statements and right click on one of the line numbers to add a breakpoint
   13. Delete the previous file and copy the new one onto the beagleboard
   14. Restart your gdb server using the terminal window in RSE using the same command as before then hit the debug button
   15. You should be able to run until you hit the breakpoint and see the console information printed on the Terminal in the RSE perspective.