Linux Workshop Setup Instructions

Abstract

This document describes the equipment, system requirements and setup for the Texas Instruments 4-day *Embedded System Design Workshop for Linux* (DaVinci/OMAP/Sitara/Integra).

Version

This document was updated August 2011 and reflect the few changes made for Embedded System Design Workshop for Linux (DaVinci/OMAP/Sitara) (Version 3.08 – Aug 2011).

Topics

Workshop Equipment (Hardware)

PC Workstation

- Intel based PC
  - Intel processor should be Core 2 Duo or greater
  - We recommend the fastest possible PC since it will be emulating a Linux PC using the VMware software listed below
  - For the same reason, we recommend at least 2GB of memory
  - PC needs to have a serial port (or a working USB to Serial Port adaptor)
- Windows XP SP2 (or higher)
  - Tested on computers running Windows XP SP2 and SP3

EVM Development Kit

- OMAP3530 EVM Board
  - The latest labs use Uboot on MMC, TFTP Linux boot, and a NFS root filesystem, therefore Flash and HDD updates are not required to be made to the board
  - Has build-in LCD Display
- EVM Power Supply

Video Source

We prefer the quality from a DVD player, but a camera that plugs into the EVM's CVBC connector would work, too.

- DVD Player (with remote)
- DVD

Cables and SD/MMC Card

- USB SD/MMC Card reader/writer
  - Workshop exercise (Lab 3) requires the student to create an MMC card used to boot the EVM.
- 2GB SD Card
  - Note, some cards don't work with EVM boards, so make sure the supplied cards are test with the EVM prior to the workshop.
- RS-232 serial cable
- Ethernet cable
- Ethernet crossover adaptor
- Composite video cable (DVD player to EVM video input)
  (single "yellow" composite cable with RCA to RCA connectors)
- Audio cable – RCA to stereo mini-jack (DVD player to EVM audio input)

Optional Hardware Items

- Router with DHCP capabilities or Ethernet switch (plus an extra Ethernet cable)
  - In the USA, we are directly connecting our EVM to the Windows PC. Our PC network interface automatically provides the crossover mechanism, therefore we do not need an external Ethernet switch or router. (Our workshop kit includes an Ethernet crossover cable, but we don't really need it since our Dell computers handle this automatically.)
  - We chose to use a direct, static Ethernet connection to minimize shipping routers/switches to each workshop location.
  - With TI SDK version 4.x (see below) we found that we cannot boot the OMAP EVM with static IP addressing. Since we still do not use hardware routers, we added DHCP server software to our Linux VMWare image (see below).
- Power strips (as needed)
  - Need (at least) 3 outlets per workstation:
    - Computer
    - EVM
    - DVD player
    - (Optional) Ethernet router/switch

Workshop Software

Windows Software

- VMware Workstation 6.54 http://www.vmware.com/products/workstation
  - All software is for Windows XP PC. The Linux code generation tools – required to build programs for the TI processors – have been installed into the Linux VMware image listed under *Workshop Lab Files* (below). VMware Workstation is required to run the VMware disc image.
  - The workshop labs have also been tested on VMware Workstation version 7.xx
  - Additionally, many folks use the workshop's VMware image with with the free VMware Player
- Tera Term – free serial port terminal software http://hp.vector.co.jp/authors/VA002416/teraterm.html
- Adobe Acrobat Reader http://get.adobe.com/reader
- Windows XP Cmd Prompt PowerToy http://go.microsoft.com/fwlink/?LinkId=211471
  (Specifically, the "Open command prompt here" powertoy.)

Highly recommended

- Notepad++ http://notepad-plus.sourceforge.net/uk/site.htm
- WinZip http://www.winzip.com
- WinRAR http://www.rarlabs.com
- Beyond Compare 3 http://www.scootersoftware.com
- MD5 Checker http://getmd5checker.com/download

TI Windows Software

TI Software (CCS)

- Code Composer Studio v4 (Tested with CCS v4.2.1.00004)
  - CGT 7.0.x was used for CCSv4 lab exercises.
  - It's possible the lab exercises will work with CCSv5 (beta), but this has not been tested
- Codec Engine 2.26 software codec_engine_2_26_01_09.tar.gz
  - This library is used with Lab 13.
  - The Codec Engine software is also installed in the VMware image below, for use when building ARM-side applications. We need it in the Windows environment for use when building/testing our DSP algorithm within CCSv4.

Note: The required DVSDK files required for building Linux applications has been pre-installed into the VMware image that's listed under the *Workshop Lab Files*.

Workshop Lab Files

TI requires that all of its computers run Windows XP. Therefore, to run the Linux code generation tools – required to build Linux programs for TI ARM processors – we utilize the VMware Workstation application to create a virtual Linux PC. required to run the VMware disc image.

- VMware Linux Image Workshop Wiki Page

  Our VMware image consists of two parts:

  1. **"vm parent" image (15 GB)**
  2. **"vm child" image (1 GB)** (note, this is contained in the "shared zip files" below)

  Hints for successfully downloading these files:

  - The VM parent image files have been zipped up with (700MB) spanning turned on. Thus, you will need to download a number of files to get the entire image.
  - All the files must be downloaded and put into the same folder. (image.zip, image.z01, image.z02, ..., image.z08)
  - After downloading the files, we recommend you verify the files using the provided .md5 signatures. We've seen the case where just one of the files had a small download error, which then prevented the unzip from occuring properly.
  - To get the smallest download size, these files were compressed using an advanced WinZip compression scheme, therefore we recommend using WinZip 10 or later.
- TTO Workshops "Shared" files
  - This workshop requires a few of additional files which are needed for use in Windows and/or Linux. We have gathered them all together in a folder called "shared".
  - As part of the workshop setup process, the "shared" folder will become visible to both the Windows and the virtual Linux environment.
  - **To make download and installation easier, we zipped up all the files needed for this workshop – except the VM Parent Image.** (As explained below, extract these to the C:\vm_images location.)

Workshop Setup Instructions

PC Setup & Login (USA Classroom)

1. **The USA classroom computers provide the following Windows login for all workstations:**

   User: student

   Password: student

2. **The computer must have a working Ethernet connection that allows it to "see" the OMAP3530 EVM board.**

   In USA classrooms, there are two network cards in each PC: one is for connecting to the TI network; the other is for connecting to the EVM. We must do this because the classroom computers sit on a firewall with limited static IP addresses, and does not provide DCHP services. A 2nd NIC provides a dedicated Ethernet port that can be configured for our workshops.

3. **Make sure the serial port is enabled and working.**

   In USA classrooms, our computers still have a dedicated RS-232 serial port, thus no special drivers need to be installed into Windows.

Windows PC Software Installation

4. **Install the following software:**
   - VMware Workstation 6.54 (or version 7.x or VMware Player)
   - Tera Term
   - Recommended Windows software (see earlier list):

     WinZip, WinRAR (works better with tar.gz files), Notepad++, Beyond Compare, MD5 Checker

5. **Configure Windows Explorer to SHOW file extensions for known types!**

   Turn off the feature that hides file extensions for known file types (turned on by default in Windows Explorer). This is important for software development, since build tools often create files with the same names, but different file extensions.

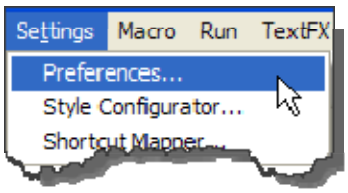
6. **Install the Windows software described in: "Recommended Windows Software".**

7. **Configure WinZip & WinRAR.**

   Associate WinZip for **zip** files and WinRAR for **tar** and **gz** files. (WinRAR is preferred for tar/gz files as it can 'explore' these file types, as opposed to just decompressing them.)

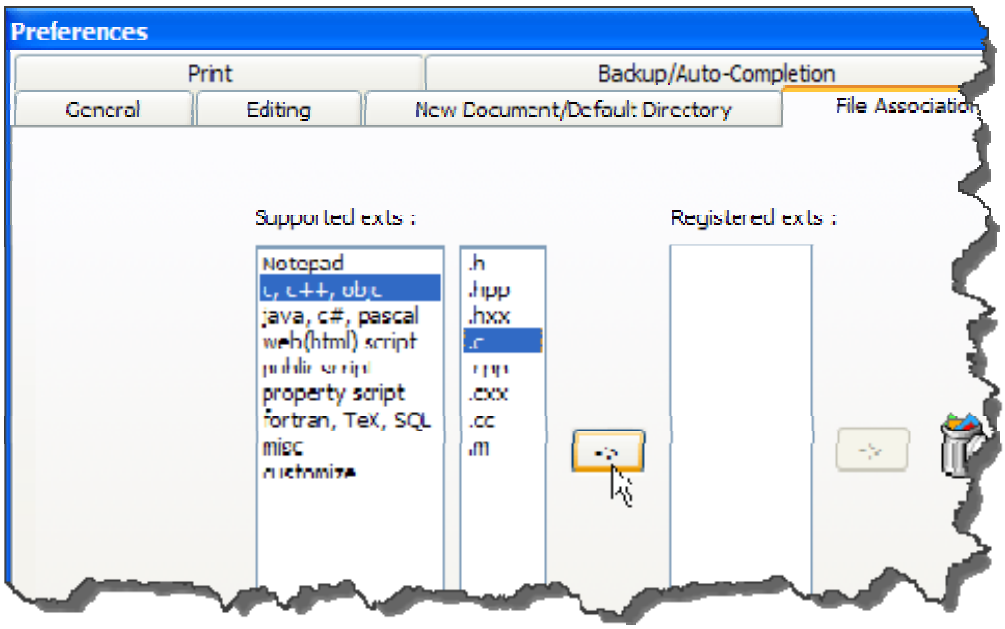
8. Configure Notepad++ as default editor.

Open *Preferences* dialog and change any n++ settings per your liking.



Though, we primarily want to change the file associations. Set source file associations (.c, .asm, .h, .cmd, .pjt, .txt, .tcf, .tci, .xdc, .xs, .bld, etc.) to open with Notepad++.

Select each filetype and then click the "->" button. For types not listed, under "customize" you can enter your own filetypes.



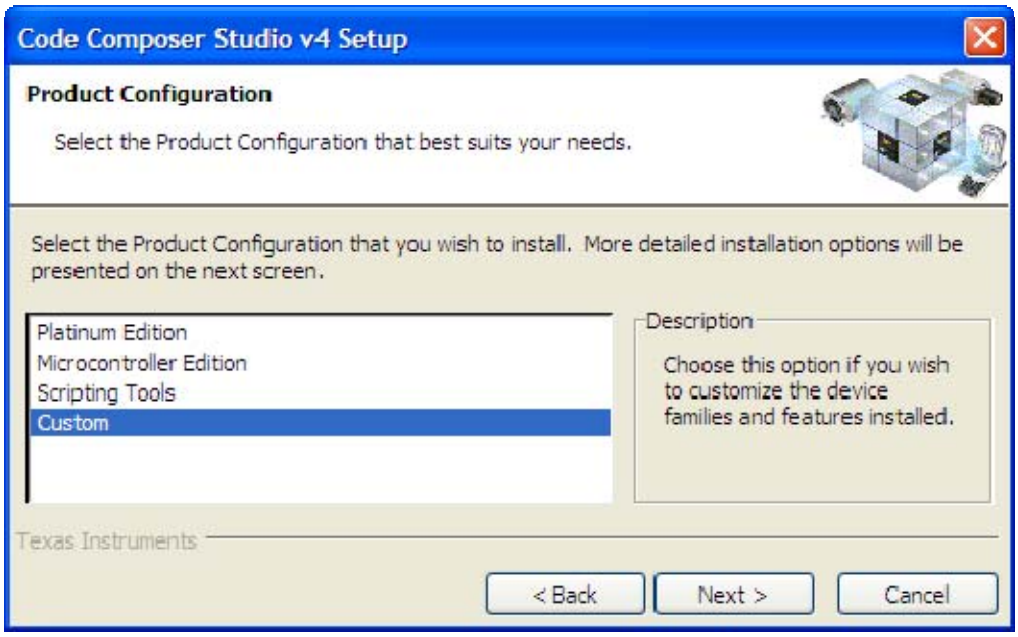
Install Code Composer Studio (CCSv4)

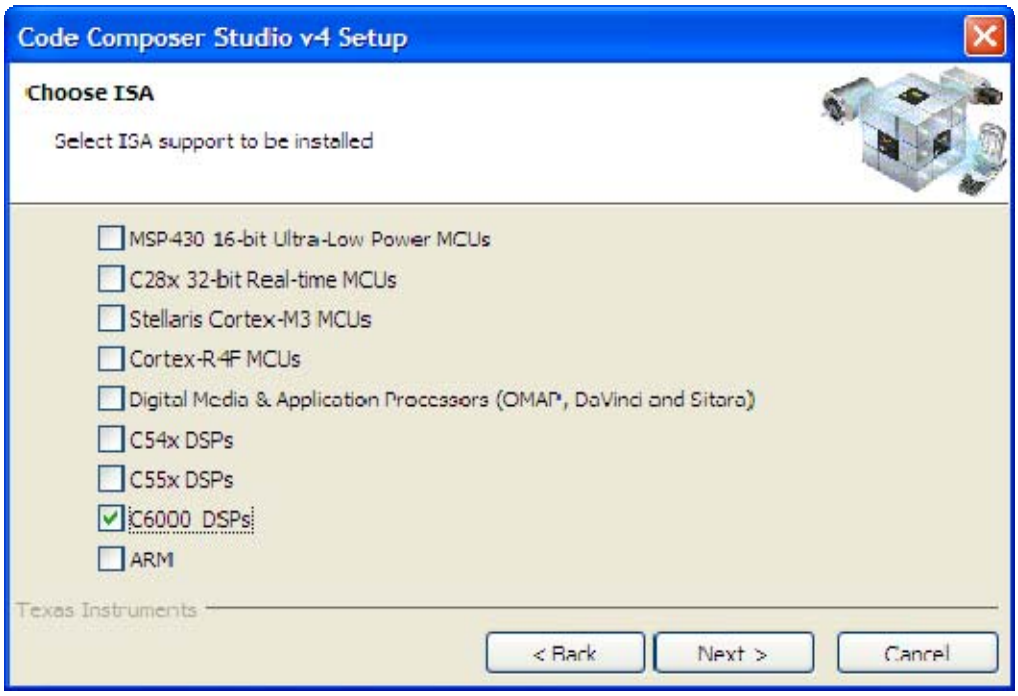
9. **Install CCSv4 (see version under TI Software requirements, listed above).**

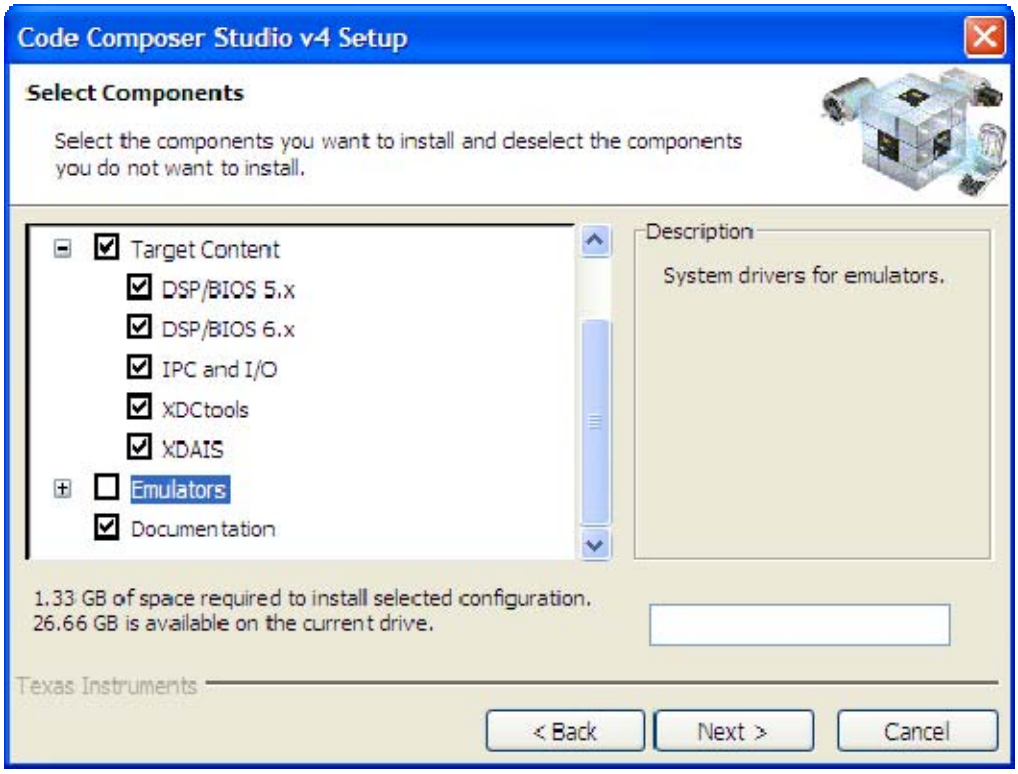
   <u>Do NOT install to the default directory.</u> Rather, install into: **`C:\TI`**

   This will create a folder `C:\TI\ccsv4` and install CCS into it.

   You only need to install for the C6000, as shown:

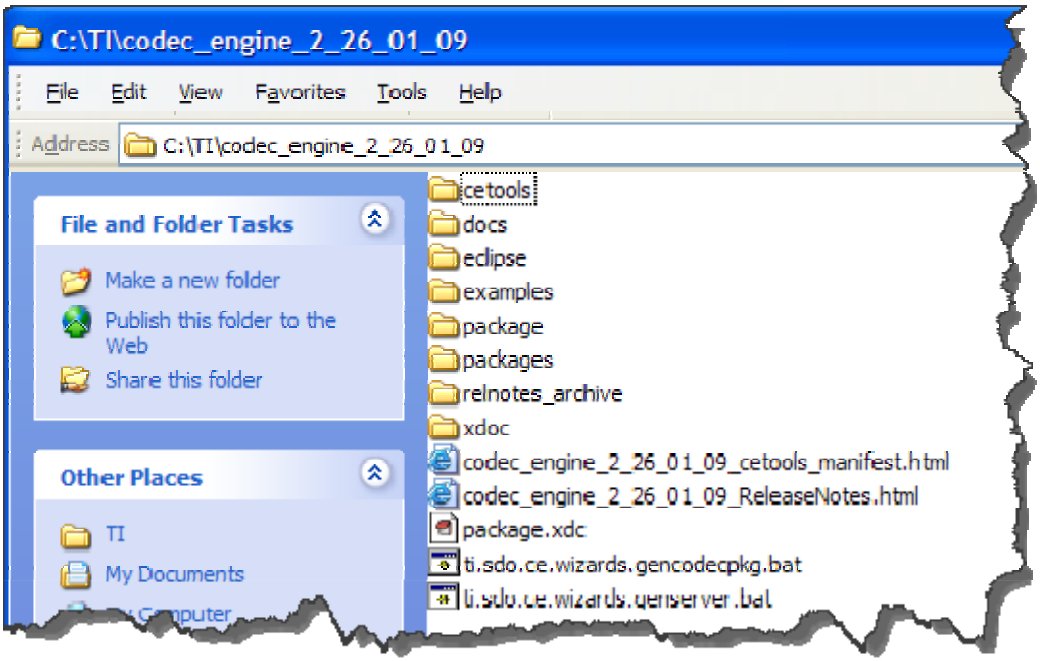






10. Install Codec Engine software.

Untar the Codec Engine files into the C:\TI directory. It should end up looking like:



Install Workshop Files

There are two workshop VMware image folders to install:

- `vm_images` (2 GB) *(Contains the VMware child image)*
- `vm_parent` (16 GB)

Why did we split the image into to folders?

We split the files into two folders to make workshop setup as quick and easy as possible. The `vm_parent` folder can be placed in a write-protected folder, partition, or drive. It should not need to be modified. (In the USA, we put this file onto a protected disc partition that users can read and executre from, but they can't modify or write to. In this way, these files will remain uncorrupted and do not need to be refreshed after every workshop.)

This makes the `vm_images` folder much smaller, allowing it to be easily replaced after every workshop. This folder contains two directories: a "shared folder" (which is discussed next) and the VMware image used in the workshop. The VMware image is quite small because it linked to (that is, it references) the files contained in the `vm_parent` folder.

11. Download the VMware parent image file.

If you haven't already downloaded the VMware parent image from the wiki, download it now and it to the workshop PC computer. (See *Workshop Lab Files* on page 5.)

It doesn't matter where this image is located. As previously noted, in the USA classrooms we place it on separate "D:\" drive partition.

12. Download and extract the vm_images ZIP file.

Rather than downloading all the remaining files individually, we suggest downloading the single .zipx file.

`Single-ZIP download for workshop files, 'shared' folder, and VMware Child image (zipx) (1.6 GB)` `Checksum (.md5)`

Extract the file to the `"C:"` drive. This should create the following directory:

```
C:\vm_images
```

Note: To decompress this file, you must use a later version of Winzip, which supports ZIPX (i.e. PPdM zip compression). This was needed to keep the download less than the wiki's 2GB limit. Unfortunately, the inclusion of CCSv5 into the Linux image pushed the file size over that limit.

13. Run the Windows workshop setup/install executable.

Run the following .exe file:

```
TTO_Linux_SOC_Workshop_labs-winfiles_omap3530_v3.06a_setup.exe
```

This installs the following files:

- Tera Term macros used by this workshop
- Desktop icon for Tera Term (configured for OMAP3530 EVM)
- *Workshop* folder – used for Lab 13

Configure VMware

14. Configure Vmware – making sure it starts cleanly for students.

Lab 3 in the workshop requires the students to setup some aspects of VMware. Even so, there are a couple items that should be done by the instructor. We recommend:

- Start VMware and close any "notification" dialogs
  (making sure to check the don't remind me again box)
- Open the correct vmware image (sets the default path to this image so students are less likely to make mistakes opening the wrong one)
- Re-enable shared folders – VM disables them after copying or cloning an image.

  After running a newly copied VMware image, you will need to re-enable the "shared" folder. VMware disables all shared folders when a VMimage is copied or moved.

Connect the Wires

15. Provide Ethernet connections between Windows PC (containing the VMware Linux image) and the EVM board.

In the USA classroom, we connect the 2nd NIC directly to the EVM board. Our PC's can do this with either a *standard* or *crossover* Ethernet cable.

It would be better if an Ethernet switch or hub was used between the PC and EVM, but we avoided this to minimize equipment shipping from location to location. The downside is that when the EVM is turned off, the "2nd network" is shut down; this may occasionally require the Red Hat Linux networking to be restarted.

16. Connect the serial port cable between the PC and EVM

17. Connect the audio and video cables.

We highly recommend you hook up and test *at least* one workstation. It is better to know in advance if a problem exists with your setup. While hooking up the rest of the workstations can be left to the students, we find it saves class time (which can be used for more important purposes) if it is done ahead of time.

18. Connect the power cable.

Device Specific

Hint: This can be accomplished using the one workstation you have setup and tested. Unplug the EVM you have been using, plug in another EVM and repeat the procedure listed in this document for all the EVM boards.

Note: For the DM6446 DVEVM labs, even if you don't plan to setup and test each workstation, you need to make sure the Uboot flash settings for each EVM board are reset to the HDD/flash boot mode. We avoided this for the OMAP35 EVM by utilizing the MMC/SD boot mode (and erasing the flash).

.

Configuring and Testing a Workstation

Note: Lab 3 (mentioned in step 19) was written with respect to USA classroom's configuration. Hopefully it will be similar to the classroom configuration used in other regions.)

19. Setup and Test (at least one) workstation.

It is a good idea to setup and test at least one workstation for each course; though, it is much safer to test all the workstations.

These are the items that should be configured and tested:

- VMware networking (bridged Ethernet)
- VMware program options (disable snapshot restoring in background)
- Test the VM shared folder – can you pass files between Linux and Windows
- Test serial port connection (ping VMware Linux image)
- Test that your SD/MMC cards work with the EVM
- Test boot with TFTP and NFS

The easiest way to test these items is to do Lab 3 from the workshop. (See the end of this document for a copy.)

Troubleshooting Suggestions

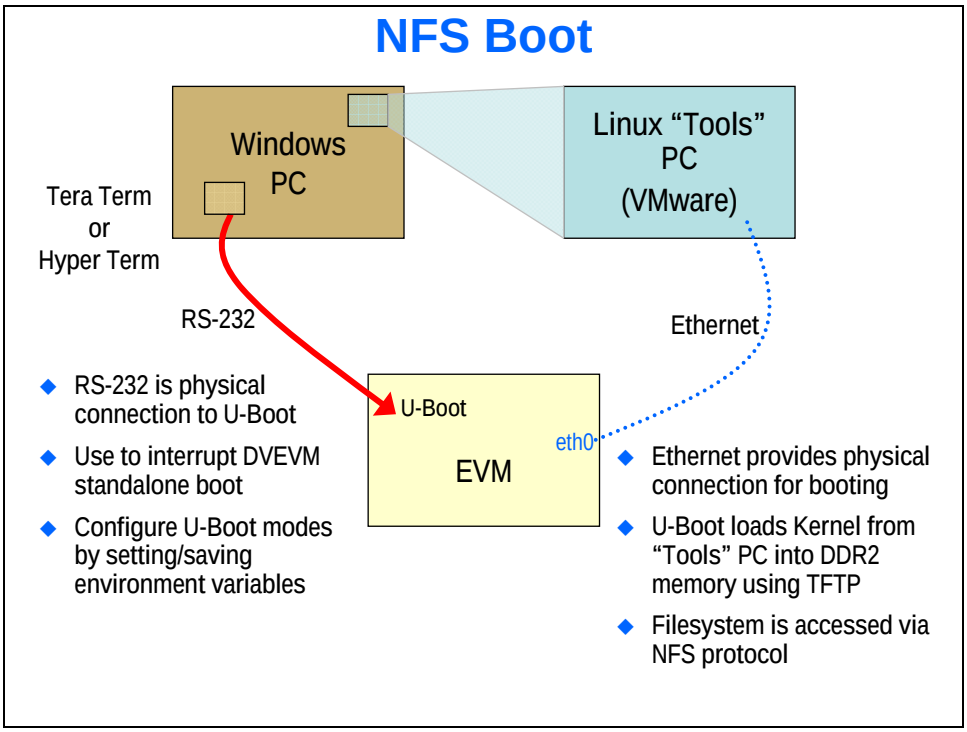
Device Specific

- On the OMAP3530 EVM, to correctly boot via TFTP/NFS from an MMC card the on-board NAND flash memory needs to be erased. You can accomplish this from within UBOOT by executing the 'erase' command:

```
nand erase
```

- See network troubleshooting suggestions in Lab 3. There's a whole page of things to check and try!

Lab3 - Experimenting with Linux and U-Boot



Most development for a Linux based target devices, such as the ARM CPU's on the OMAP/Sitara/DaVinci, is done on Linux-based host machines. Developers with Linux PCs can therefore work directly in this environment, but authors using Windows based PCs need either to obtain a new PC running Linux, or employ software that can simulate the Linux environment on top of Windows. In this workshop, VMware is used to create a 'virtual machine' on a windows PC, inside which the Ubuntu operating system can run. In this portion of the lab, the steps to configure Ubuntu on VMware will be implemented. In this lab, the following steps will be taken to set up the software development environment:

Chapter Outline

Lab03a – Start/Configure VMware and Ubuntu Linux

VMware

20. Launch VMware.

On the Windows desktop, **double click** the **VMware** icon.

21. Open the TTO workshop VMware image.

In the VMware Workstation window **Home tab**,

`Click on the Open Existing VM or Team Icon`

Open the VMware image file (the name you see might be similar but not exact):

`C:\vm_images\tto_vm_child_image_(v3.08)\tto_vm_child_image.vmx`

Notes:

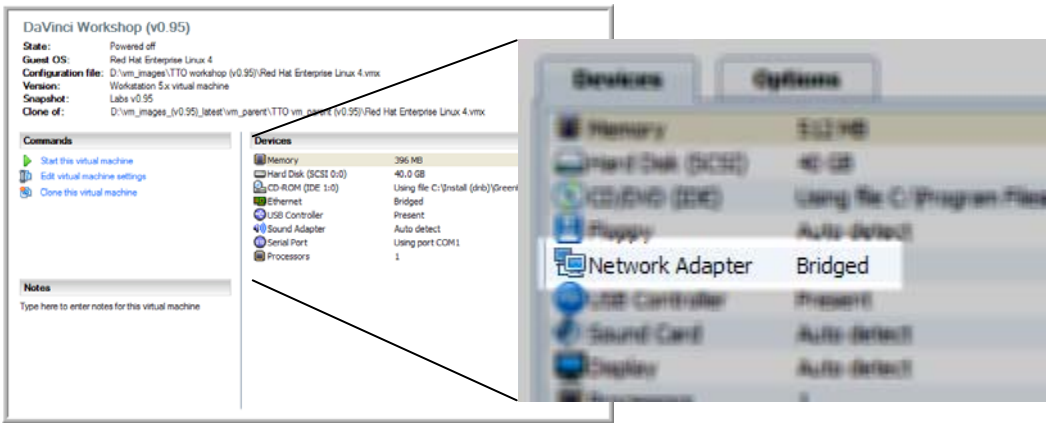
- If your instructor has already started VMware for you, then you may skip this step.
- VM image version v3.08 was current at the time of this writing.
- In USA classrooms, the VMware image is broken into two parts:
  1. *Child* image (~1.8GB) (C:\vm_images\tto_vm_child_image_(v3.08)\tto_vm_child_image.vmx)
  2. *Parent* image (~16GB) (E:\vm_parent\TTO_vm_parent _(v3.03)

  The child image, specified in this step, depends upon the parent in order to work. Breaking the image into two parts allows us to re-image the C:\ drive being required to reload the entire 18GB for each class.

22. **Verify the Linux networking options are set to 'bridged' mode.**

This option tells VMware to access the network and obtain its own IP address (other choices involve the Windows PC acting as a router). If not set to 'bridged'

If you have opened VMware application and the TTO image, you should see the Ethernet setting in the middle of the VMware screen as shown here:



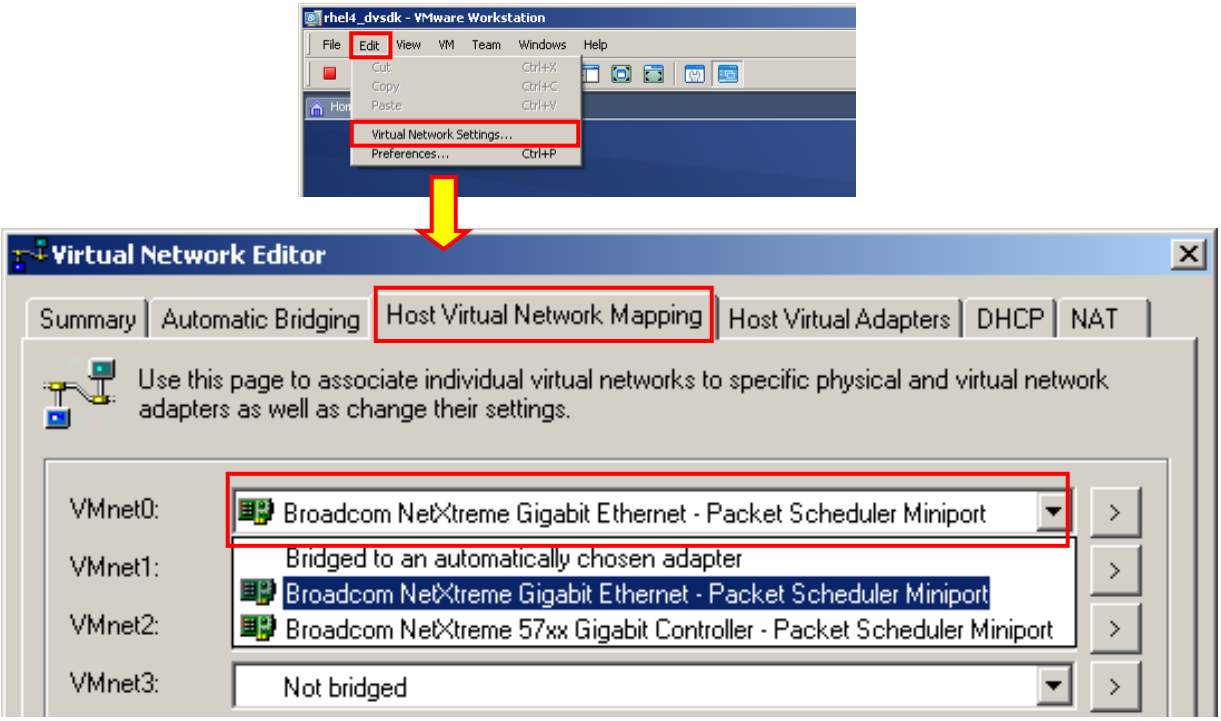
If you happened to get a little ahead of our instructions and already started the VMware image (which we do in step 25), the easiest way to see this is in the status bar. Just hover over the Ethernet board icon and read the popup message:

Note: If you are using the VMware player, this information is easily found via the top toolbar. In USA classrooms, we use the full version of VMware, though, as opposed to the limited Player version.

23. **Define which of the Ethernet ports on the PC Linux we will use.**

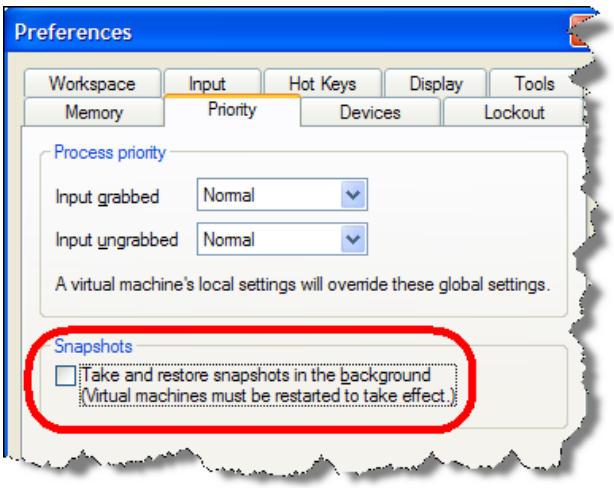
(Note, this step is required for USA TI classrooms, but may not be needed when using laptops within the USA or for other non-USA locations. Please check with your instructor if you are not sure if this applies to you.)

From the VMware Workstation menus, select **Edit | Virtual Network Editor…** In the Virtual Network Editor dialog box that appears, go to the **Host Virtual Network Mapping** tab. In the drop box for **VMnet0**, select the **Broadcom NetXtreme Gigabit Ethernet Packet Scheduler Miniport** adaptor, as depicted below:



24. To improve system speed, disable the VMware snapshot feature.

Under **Edit | Preferences**, go to the **Priority** tab, and **uncheck** the **Snapshots** feature. Close the window by clicking on the **OK** button. (If using the VM Player, this option does not apply to you.)



Note:

Your instructor may already have booted your Ubuntu image (in VMware) and left it hibernated (paused). If so, steps 25& 0 might act slightly different.

25. Start Ubuntu Linux.

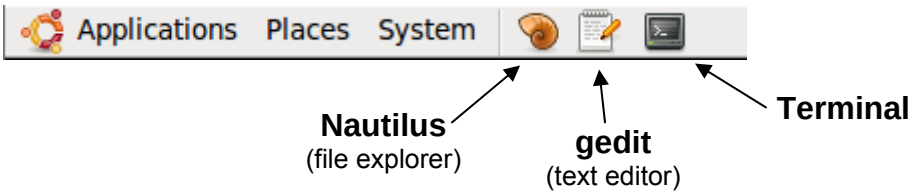
Click on the green '**Play**' arrow  in the icon bar near the top of the VMware window. (Another way to start the Linux session is to select Start the Virtual Machine in the Commands area). Wait for the boot process to complete (which may take between 2-5 minutes), as indicated by the appearance of the **Log On dialog box**. (If using VM Player, the image is automatically started when opening the VMware Image file.)

Ubuntu will automatically log you into Linux with a user account. At this point, you will simply see a blank desktop and you can move on to the next step.

FYI – Ubuntu automagically logged you into the following account – *no login required by you at this time:*

Ubuntu Userid:	**user**
Password:	*none required*

26. Open a terminal window.



The easiest way to open the terminal is to click it's icon on the panel toolbar. You can also find it on the "Applications" menu, but we've placed icons to the three most-used tools onto the toolbar panel.

Lab03b – Install Workshop Lab Files (for your board)

We have installed the appropriate software for your EVM board.

That is, we have worked thru the Getting Started Guides (GSG) for each of the boards (OMAP3530 and AM3517) into the same VMware image, because they both use the same DVSDK/SDK (software development kit) and version of community Linux.

Since the DM6446 uses a different DVSDK, we chose to install its software libraries (and MontaVista Linux) into a separate VMware image. *(See v2.10 of lab materials/files for this target.)*

In this part, you will install the workshop labs/solutions files per the board you have chosen to work on during class. Additionally, we will configure/verify a couple of environment settings.

Installing Workshop *Labs* and *Solutions* Files

27. Verify the `shared` folder is enabled.

Let's try simply listing the files in the shared folder. If there aren't any files, we may need to enable this VMware feature.

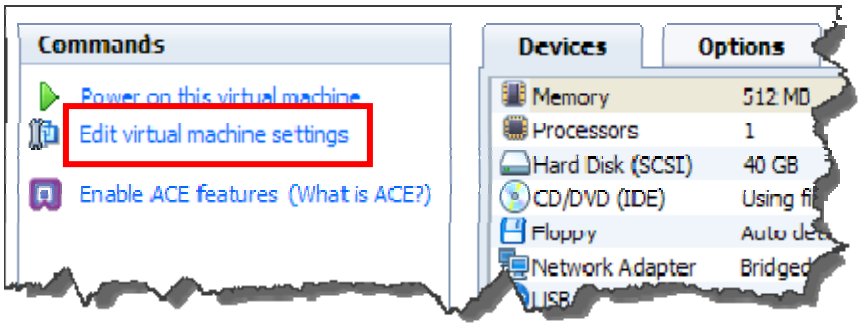
```
ls -l  /mnt/hgfs/shared
```

If this doesn't work, shared folders are not enabled. Continue with the next step to enable shared folders.

28. If needed, enable shared folders.

If VMware Workstation is running (and it probably is, at this point), go to "options" view by clicking on the **Options** toolbar button:

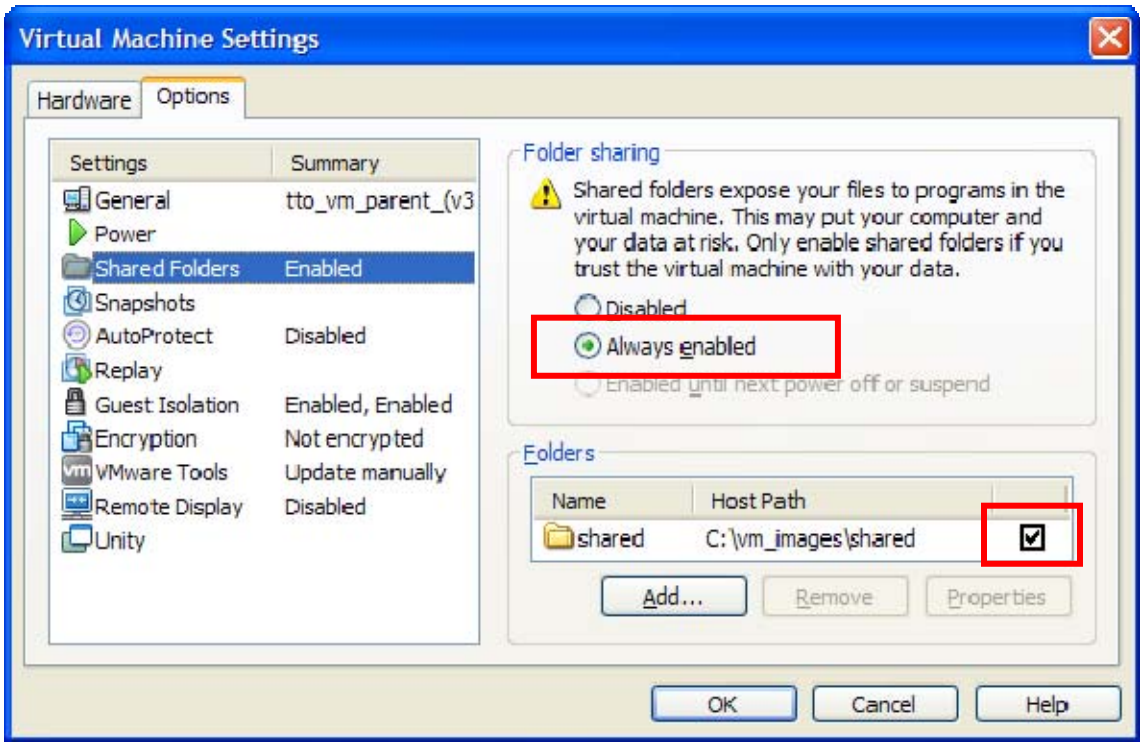
Click on **Edit Virtual machine settings:**



And then …

When finished enabling shared folders, simply click the "Console View" button in VMware to get back to the command line.

Make sure that **Shared Folders** are *Always enabled*:



29. Copy lab files from Windows/VMware shared directory.

Device Specific

To keep things simple, for the OMAP3530 and AM3517 VMware image, everything but the lab files have been installed. Rather than putting lab files for both target boards in the user folder, we have provided you two tar files.

```
cd /home/user
cp /mnt/hgfs/shared/TTO_Linux_SOC_Workshop_labs_omap35_v3.xx.tar.gz .
```

Options:

- For the AM3517 choose: `TTO_Linux_SOC_Workshop_labs_am3517_v3.xx.tar.gz`
- Rather than seeing a file with `v3.xx`, choose the latest revision available; e.g., `v3.08`.

30. Untar the lab files into the `/home/user` folder.

In the steps below, make sure you use the file you copied in the previous step.

```
cd /home/user
tar –xzf TTO_Linux_SOC_Workshop_labs_omap35_v3.xx.tar.gz
```

After unzipping, you should have two new folders in your `/home/user` folder. If not, please consult with your instructor.

```
/home/user/labs
/home/user/solutions
```

31. Verify you have installed the correct files for your EVM platform.

Device Specific

(You can skip this step if you are following the DM6446 labs.)

Check that the readme file exists in your new labs (and/or solutions) folder. We use the readme file to confirm the platform supported – along with the workshop labs/solutions version number.

`/home/user/labs`

`Readme_omap35_labs_v3.xx.txt`

or **`Readme_am3517_labs_v3.xx.txt`**

32. Add symbolic link to `targetfs` directory. *(You can skip this step if you are doing the DM6446 labs.)*

Finally, we need to add a Linux symbolic link for our **`targetfs`** directory.

Device Specific

```
ln -s /home/user/psp_rebuild_omap3/linux_filesys /home/user/targetfs
```

or

```
ln -s /home/user/psp_rebuild_am3517/linux_filesys /home/user/targetfs
```

This Linux command (small LN) creates a symbolic link, similar in some ways to a Windows shortcut. With this link, we can now refer to the `/home/user/targetfs` directory in our workshop instructions and the correct folders/files will be referenced on each of your systems, no matter which EVM you are using.

This is also the directory we are "exporting" (i.e. network sharing). We already set this up for you in Linux by editing the /etc/exports file. This was required because since this is the folder used as the *nfspath* – that is, we will use this folder (via the network) as the *root filesystem* for our EVM.

Installing kernel modules to the *targetfs*

33. Install the kernel modules and `loadmodules.sh` script to the target filesystem.

We have conveniently placed the kernel modules and scripts into the lab00 folder. All you need to do is run the install script located in that folder.

```
cd /home/user/labs/lab00_install_scripts
./install.sh
```

This will copy the files contained in this folder over to our workshop directory in the target filesystem (`/home/user/targetfs/opt/workshop`). Later on we'll discuss what these files are used for; for now, we just want to copy them into place so they'll be there when we need them.

Lab03c – Image SD/MMC card (to boot EVM)

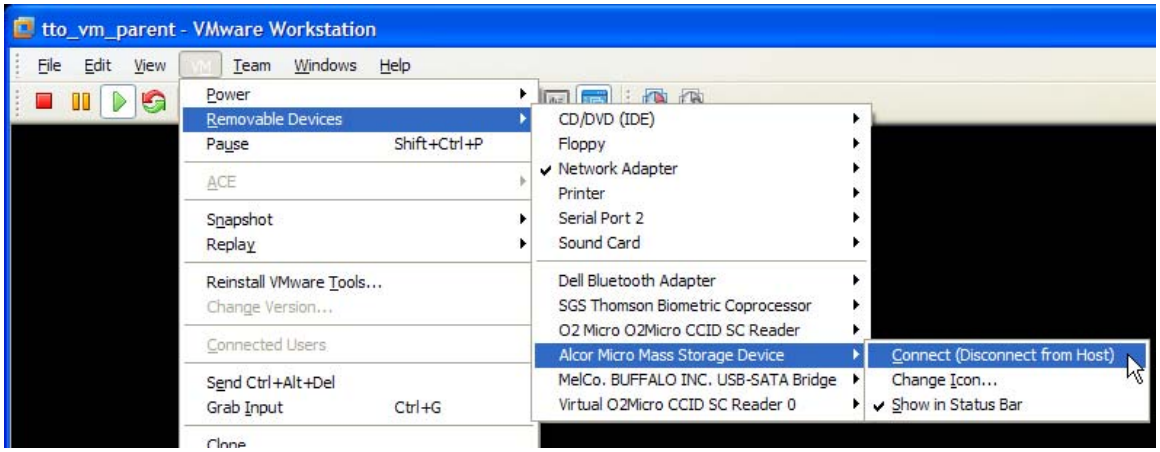
34. **Plug USB Flash SD/MMC Card reader into a USB port on your computer.**

    You may see a dialogue box talking about "Removable Devices" – just click OK and continue.

35. **Connect the SD/MMC flash card reader to the Ubuntu virtual machine.**

    If USB Flash Card reader is mapped to Windows host, select:

    `VM → Removable Devices → <Your Flash Reader> → Connect…`



Note: Your SD/MMC card reader may show up as a slightly different name, depending upon the brand of reader you are using

36. **Open a terminal window in Ubuntu** (if one is not already open)**.**

37. **Move to the Lab03a directory.**

```
cd ~/labs/lab03_build_sd
```

38. Determine SCSI device node for USB SD card reader

```
(user@ubuntu) #  sudo sg_map -i
When prompted for sudo password, (press enter)
```

You should see a table similar to the following:

```
/dev/sg0  /dev/scd0  NECVMWar  VMware IDE CDR10  1.00
/dev/sg1  /dev/sda   VMware,   VMware Virtual S  1.0
/dev/sg2  /dev/sdb   USB 2.0   SD/MMC Reader     1.0
```

Depending on the SD/MMC Reader used, it may appear differently, but will likely be the last device on the list.

Write down the Linux device node (i.e. virtual file name) for the card reader:

Your device node: _____ (most likely, `/dev/sdb`)

39. Insert a 2GB SD/MMC card into the USB Flash reader (if not already done)**.**

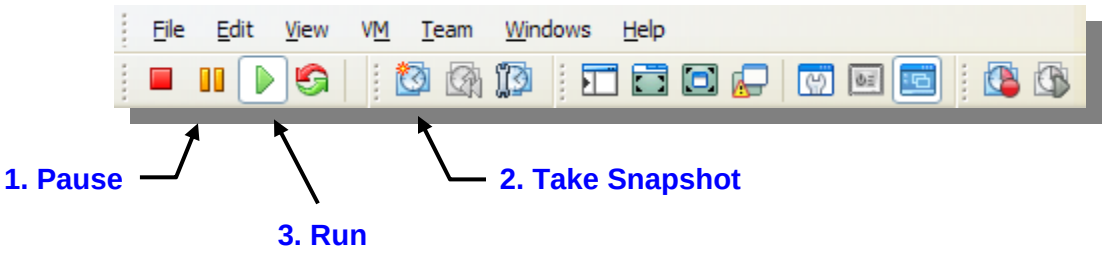
Caution

Read the following step and comments very carefully, specifying the wrong `/dev/sdx` device node could cause permanent damage to your system!

40. Take a VMware snapshot. (*Only full version of VMware Workstation supports snapshots.*)

Because this step could erase the wrong drive in your system, let's make a snapshot copy of our virtual hard drive. This can be done many ways, but we suggest this simple 3-step procedure – which uses three different VM toolbar buttons:

1. Pause your Linux VMware PC.
2. Take a snapshot.
3. Un-pause (that is, Run) your Linux VM, again.



41. Execute the `build_sd.sh` script.

Run the build script using the device node from step 38 (page 3-22). If prompted for a sudo password, simply press enter (blank password).

```
(user@ubuntu):  sudo SCSI_DEV=/dev/sdb ./build_boot_sd.sh
```

When asked to "confirm", press "y" and [ENTER].

It should take less than a minute for the script to complete. The script automates these steps:

- Un-mounts partitions (if any) that Ubuntu automatically mounts to the desktop
- Reformats and formats the SD/MMC card for two partitions
  (though we'll only use one, for now)
- Temporarily mounts new partitions and copies three files onto the 1st partition:

```
MLO          (X-loader – 2nd level bootloader)
u-boot.bin   (uboot – 3rd level Linux bootloader)
uImage       (Linux kernel)
```

In the next part of the lab, we'll use the MMC card to boot the EVM.

Lab03d – Talking to the EVM

42. Start TeraTerm.

On the Windows desktop, **double click** on the **TeraTerm** icon.
The TeraTerm serial configuration file dvevm.ini, in the TeraTerm program folder has already been set up with the following necessary configuration states:

```
Bits per Second: 115200
Data Bits: 8
Parity: None
Stop Bits: 1
Flow Control: None
```

43. Insert the SD/MMC card into the EVM.

If you haven't already done so, remove the SD/MMC card you formatted in step 41 from the card reader.

```
Insert the card in the EVM's SD/MMC card slot
```

The card should go into the slot "label up" – SD card "pins" down. On new boards, the slot is tight, so you need to make sure and line it up very straight as you slide the card into it.

44. Connect RS-232 serial cable.

If not already done, please connect the serial cable. (If unsure how to do this, please as your instructor (or refer the EVM Quick Start Guide).

```
Connect RS-232 cable between the EVM and PC RS-232 port
Note: For OMAP3530 EVM, please use the UART1/2 connector.
```

Device Specific

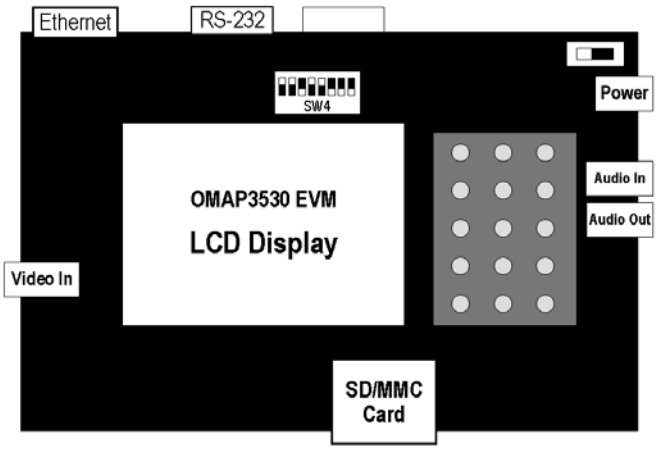
45. Verify EVM Hardware Configuration

- Is the EVM powered off?
- Verify the switch settings (for proper booting) of the EVM - where does board find MLO and uboot.bin?

OMAP3530 EVM switch S4

```
SD/MMC card:      0010 0111
On-board NAND:    xxxx xxxx
```

(AM3517 switch settings continued on next page.)

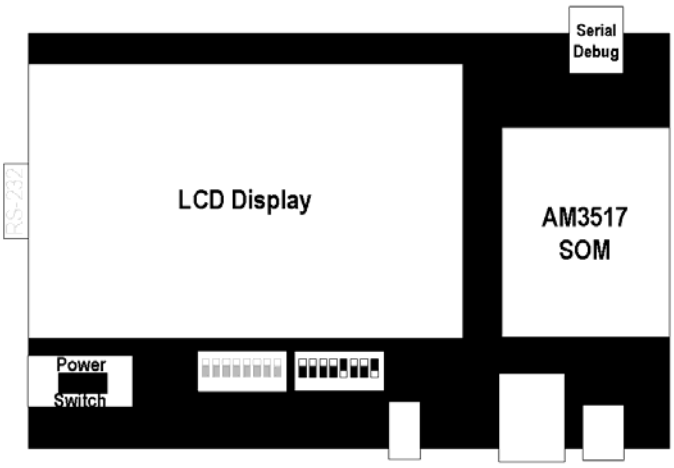


AM3517 EVM switch S7

```
SD/MMC card:     0000 1001
```

Setting the first and fourth switches on, while the others are off, tells the board to boot using the MMC card.

```
sw 7-1: on
sw 7-2: off
sw 7-3: off
sw 7-4: on
sw 7-5: off
sw 7-6: off
sw 7-7: off
sw 7-8: off
```



These switches modify the boot mode pins on the AM3517, which are used by the ROM bootloader (1st stage) to use the XLOADER (2nd stage bootloader) found on the first partition of the MMC card. If all switches are off, the device will boot using the XLOADER found in the EVM's onboard NAND flash.

To learn more about the switches (and configuration) of the AM3517 board, visit:

http://processors.wiki.ti.com/index.php/GSG:_AM35x_EVM_Hardware_Setup

46. Start the EVM – by plugging in the Power cable (or toggling the switch next to power cord, if there is one).

Power on the EVM board and press any key to interrupt U-Boot's boot sequence.

```
Press any key    (to stop Linux from booting)
```

At this point, the EVM U-Boot terminal prompt (`DaVinci EVM#, OMAP3#, AM3517#`) should be visible in the TeraTerm session window.

(Troubleshooting Note: if the OMAP35 EVM does not seem be booting from the SD/MMC card, it's possible that it is ignoring the card and booting from the NAND flash instead. We have erased the NAND memory on the TI classroom boards to prevent this from occuring.. You can accomplish this from within UBOOT by executing the 'erase' command (`nand erase`).

In a few minutes we'll setup U-boot and get the board running …

Lab03e – Verify Networking and Record IP Addresses

Connecting to the Network

47. Make sure the Ethernet cable is connected between your EVM and the PC where you're running VMware.

If you're direct connecting the VMware image to the target EVM, then make sure the SD/MMC card you just programmed is inserted into the EVM and then power-on the EVM board (we don't care what it does at this point – that will be handled in the next section).

On the other hand, if you're using a switch or router, simply make sure that the swtch is up-and-running and connected to the EVM and PC.

48. Record the Windows Ethernet address.

This information will be used to test the Linux Ethernet connection in the next step. In the Windows **system tray** (right side of the Windows task bar) **double click** on the Local Area Connection 2 icon: 

From the **Support tab** of the dialog box that popped up, write the noted values below. Close the window when done recording this value.

IP Address _____

Note, If there are two wired LAN icons in the Windows taskbar, you should choose the one with the IP address: 192.168.1.39

49. Determine the Ubuntu Ethernet address.

You must have the Ethernet cable plugged in to the EVM and the board powered on or you will get an error during this step. (You should have connected the Ethernet cable in step 47.)

In the terminal window, run **ifconfig** by typing: **/sbin/ifconfig** ↵ and transcribe the IP address below. (Alternatively, we've set our $PATH statement so that you can just type *ifconfig*.)

IP Address _____ (Note, it will be called "inet" in the Linux response)

50. Test the Linux Ethernet port:

Ping the Windows Ethernet port to verify that both are working. In the terminal, type:

```
ping <IP_Address>
```

Where *IP_address* is the value recorded in step 48 above. The response should look like:

```
[user@localhost ~]$ ping 192.168.1.39
PING 192.168.1.39 (192.168.1.39) 56(84) bytes of data.
64 bytes from 192.168.1.39: icmp_seq=0 ttl=128 time=2.00 ms
64 bytes from 192.168.1.39: icmp_seq=1 ttl=128 time=0.675 ms
64 bytes from 192.168.1.39: icmp_seq=2 ttl=128 time=0.800 ms
```

In Linux, you need to halt the ping command using:

```
<Ctrl> C to halt the pinging
```

These are the IP addresses we plan to use in this workshop:

Windows PC:	`192.168.1.39`
Ubuntu Linux:	`192.168.1.1`
EVM target:	~~`192.168.1.41`~~ `dynamically set`

Network Troubleshooting

Skip this is everything is working…

Normally, you won't need to use any of these hints, but here's a few ideas we've run across when we cannot connect the Ubuntu VMware image and Windows.

- Don't waste your time debugging the Ubuntu to EVM connection if you can't even connect between Ubuntu and Windows (which are on the same physical machine).
- Make sure that networking is alive – in most cases, since we're direct connecting from our PC to the EVM, this means the EVM needs to be turned on with Uboot running. Of course, if you're using an Ethernet switch, then it needs to be turned on. Also, you may want to verify that the cable connection is working by observing the networking port lights at both ends.
- Make sure VMware's networking is turned on and in "bridged" mode (step 22).
- If your machine has more than one network card (i.e. NIC), make sure VMware is configured to use the correct one. We configured this in step 23, *"Define which of the Ethernet ports on the PC Linux we will use."*
- Using VMware and Ubuntu, we have occassionally run into a weird side-effect where Linux caches MAC addresses – but VMware changes the MAC addresses to avoid potential conflicts when copying or moving VMware images. This seems to be exacerbated when using Ubuntu's easy "Network Connections" feature. Solving this problem is a three step process:
  a. Delete any network connections from Ubuntu's "Network Connections".

     `System > Preferences > Network Connections`

  b. Run a small script we've added to our VMware image:

     `fixmac.sh`

  c. Reboot Ubuntu Linux.
- Ok, we're running out of ideas. So, make sure that your various Windows or Linux network settings are causing problems – things like firewalls, antivirus programs, proxy settings. We eliminated this on classroom desktops/laptops before class, but if you aren't using our PC's…
- As a side note, even if you connect the VMware image to internet, you may not be able to access it. (To minimize cabling and hardware, we don't provide this type of access to the VMware images in our classes.) If you must have access, you need to run `sudo dhclient`, to get an IP address from the network. Again, you also must deal with firewalls, proxy's and such.

Lab03f – Configure U-Boot and Boot the EVM

51. Return to Windows and TeraTerm.

Since VMware implements a complete virtual PC when the cursor is within its borders, it is necessary to move the cursor outside the VMware frame so that the use of **Alt + Tab** will invoke the underlying Windows OS and allow control to pass from the VMware application to another Windows program. Then, hold down the Alt key and repeatedly pressing Tab until the **TeraTerm** application is selected.
Release the Alt key to complete the selection.

Based on where we left things earlier in the lab, you should be at the U-Boot prompt. If this is not the case, power-cycle the board and then stop U-Boot from booting into Linux by hitting any key.

Device Specific

52. Run the TeraTerm macro to setup the EVM's U-Boot mode.

From TeraTerm, select **Control | Macro**. From directory **C:\Program Files\TTERMPRO** select the file associated to your board:

```
DM6446 DVEVM:    tto_uboot_setup.ttl
OMAP3530 EVM:    tto_uboot_setup_3530.ttl
AM3517 EVM:      tto_uboot_setup_3517.ttl
```

If the macro pauses, simply hit [ENTER] in the terminal window to continue with the questions below:

As the macro runs, make the following selections:

- Use Default NFS Server IP Address: 192.168.1.1 **Yes**
- Boot Static or Dynamic? [Yes= Dynamic (dhcp), No=Static)] **Yes**
- Root Filesystem from NFS or MMC? (Yes=NFS, No=MMC) **Yes**
- Use default NFS path? (/home/user/targetfs) **Yes**
- Kernel from TFTP or Flash/MMC? (Yes=TFTP, No=Flash/MMC) **Yes**
- For TFTP boot, use the default Kernel image filename? **Yes**
- Save bootargs? **Yes**
- Boot Linux now? (No, we'll do this manually) **No**

53. Test network connection from EVM to Ubuntu VMware image.

This is a good thing to check, since we plan to boot the EVM across the network – that is, we plan to get the root filesystem (and maybe the Linux Kernel) for the EVM from our Ubuntu Linux VM image.

Run the **ping** command from Uboot.:

```
ping 192.168.1.1
```

It should respond with: `Connection is alive`

54. **Examine the EVM's Linux environment.**

    The **printenv** ↵. reports the current state of the U-Boot variables. You should be able to see the changes we made with our interactive TeraTerm script.

55. **Save the new U-Boot settings.**

    Changes to the environment must be saved to the Flash to remain active after power-cycling the EVM hardware. This is done automatically by the macro when you answer **Yes** to the 'save bootargs' question.

    To manually preserve the bootargs, type:

    save ↵

56. *Take Home exercise…*

    Review the macro by opening the file in any text editor. While not commented in detail its code should be easy to understand if one knows the U-Boot options in general.

57. **Boot / Reboot the EVM.**

    Power-cycle the EVM or type boot ↵ to restart the EVM with the new environment settings. When boot completes (you can watch it in Tera Term – should take a few minutes), **log in as root user**; no password is needed. *Note: if during bootup "kernel panic" is reported, ask the instructor for assistance.*

    Your Windows terminal (i.e. Tera Term) is now connected to the "target" Linux running on the EVM's ARM processor.

Sidebar

It is common practice to log into a host Linux PC as a user (i.e. not as the root user). Conversely, it is also common practice to log into a development board, like the EVM using the root user. In embedded applications, there often only exists a single user (root).

58. Verify shared file system between Ubuntu and EVM.

Since any file change to the root directory of our EVM board will be reflected in Ubuntu Linux, let's give it a try by creating a new file (or updating its timestamp) using the Linux "touch" command.

From Tera Term (which is now logged into the EVM board):

```
root@omap3evm:~# cd  /                          moves you to root
root@omap3evm:~# touch putfileatroot.txt  create a new empty file at root
```

Now, let's look for this file on the NFS source directory; that is, in the target filesystem on our Ubuntu PC. To do this, list the files of the target filesystem **from the Ubuntu terminal session** (note: be careful to be in the correct window, as there are two that can be easily mistaken for each other) you started earlier:

```
[user@localhost user]# cd /home/user/targetfs
[user@localhost user]# ls -la
```

You should see the *putfileatroot.txt* in your listing, with the current date and time stamp (you could always try the Linux *date* command if you'd like to change it to your time zone). Note, you can see the same directory (and file) from both environments. Similarly, when we create new app's within Ubuntu Linux, if they are created (or copied to) our target filesystem folder, they're immediately available at our NFS mounted EVM target.

Review

To summarize, the root path of the EVM is set to a path inside the User's home directory. Fill in the box below indicating the path within Ubuntu Linux where the EVM board's root path is associated.

EVM Board "Target"		Ubuntu Linux "VMware Image"
/	=	

How did this association get made? _____

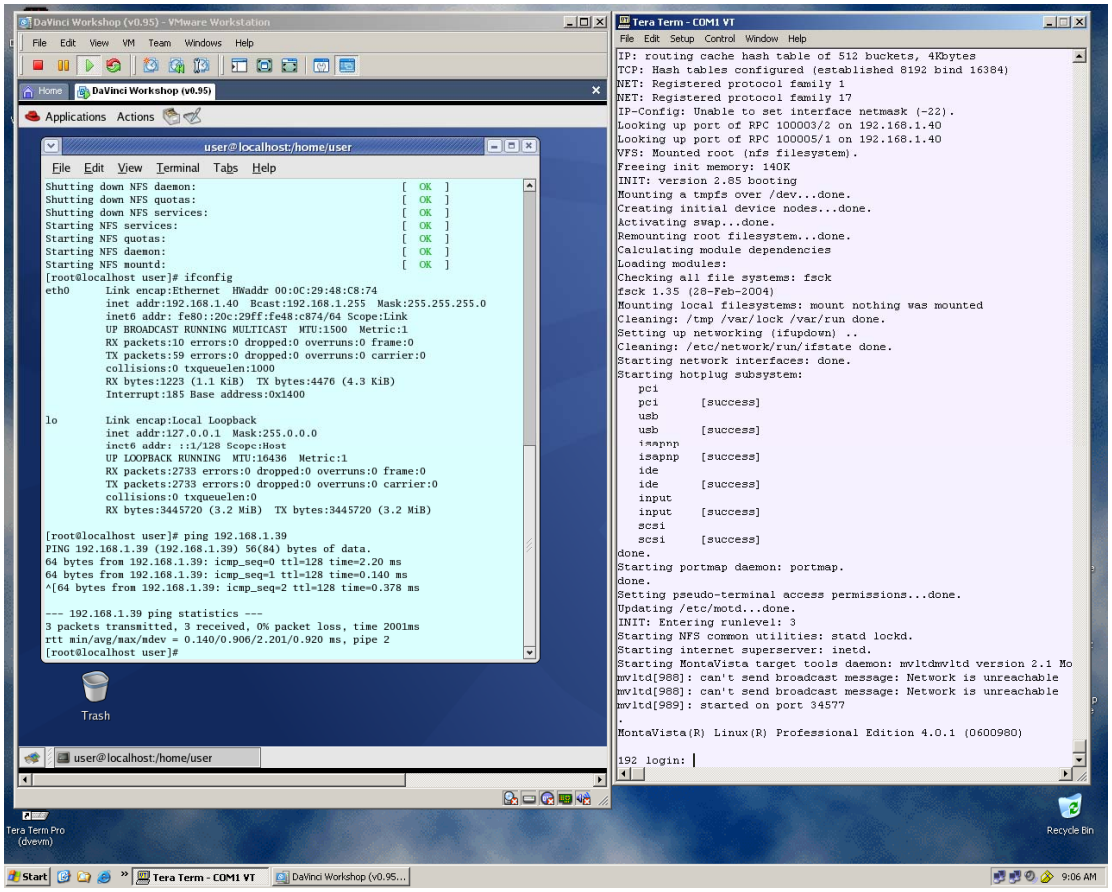
_____

What is the advantage using an NFS (networked) mounted filesystem versus using the hard drive (or flash drive) built into the DaVinci board? _____

_____

(Optional) Lab03g – Try Other Boot & VM Options

a. Arrange your Desktop windows to show both terminals side-by-side. This might make it easier to keep from confusing one terminal versus another. *(Doesn't really work well on small laptop computer screens, but works great with larger monitors.)*



Device Specific

b. Try booting up the board with other combinations of options:

- If you have a router, you could try dynamic IP addresses (vs Static IP).
- On DM6446 DVEVM, try … Kernel/Root filesystem: tftp/nfs, tftp/hdd, flash/hdd, flash/nfs. (Note, though, you will need to update the flash on the board, first.)
- On the OMAP35 or AM35 EVM's:

		MLO (i.e. xloader)	u-boot	Kernel	Filesystem	Comments
✓	1	mmc	mmc	tftp	nfs	This was lab Lab03e.
	2	mmc	mmc	mmc	nfs	Re-run TeraTerm setup macro to choose these options.
	3	flash	flash	tftp	nfs	Need to program the flash first, see Software Dev'l Guide instructions. Then, re-run the TeraTerm setup macro.
	4	flash	flash	flash	nfs	
	5	mmc	mmc	mmc	mmc	Later we'll examine copying the filesystem to SD/MMC.