# AM3517 Half-Day Workshop

## *Lab Exercises Setup Procedure*


**Revision 1.1**
**April 2010**

*Technical Training Organization (TTO)*

# Notice



Training Technical Organization
Texas Instruments, Incorporated
7834 Churchill Way, MS 3984
Dallas, TX 75251-1903
(972) 917-3889

# Training Materials Wiki

The training materials used for this workshop can be found at:

http://processors.wiki.ti.com/index.php/AM3517_On-line_Workshop

**Note:** Throughout this lab handouts document we use the phrase *AM3517 EVM* to denote the target board we are working with. In fact, most of you will be working with the *AM3517 Experimenter Board* with the optional LCD module attached. The *Experimenter Board* is actually a subset of the *Evaluation Module* (*EVM*); the *EVM* kit contains an additional board which provides additional ports, connectors, and options. Either kit may be used to complete these exercises.

# Table of Contents

# Lab Hardware and Software Requirements

## Hardware (all items are 1 per lab station)

- Laptop PC:
  - Must have at least 40 GB harddrive
  - Windows XP operating system installed
  - 1 RJ-45 Ethernet port
  - 1 USB (A-connector) port
  - Power supply
- Logic PD Zoom AM3517 Experimenter Kit
  - http://logicpd.com/products/development-kits/zoom-am3517-experimenter-kit
  - Model Number: SDK-XAM3517-10-256512R
  - (AM3517 EVM can be used in place of the Experimenter Kit as it's the same kit with one additional interface board.)
- Logic PD 4.3" WQVGA Display Kit
  - http://www.logicpd.com/products/display-kits/43-wqvga-display-kit
  - Model Number: LCD-4.3-WQVGA-10R
- Ethernet Crossover Cable
  - (Not included with Zoom Experimenter Kit)
- USB A to B cable
  - (Not included with Zoom Experimenter Kit)
  - Note this must be a full-size USB B type connector, not a mini-B connector
- 2 GB SD Flash card
  - (Not included with Zoom Experimenter Kit)

## PC Software Requirements

- Windows XP operating system
    - (also specified in hardware section under Laptop PC)
- Windows FTDI driver (CDM 2.06.00 WHQL Certified)
    - Download from workshop wiki: http://software-dl.ti.com/trainingTTO/trainingTTO_public_sw/am3517_odw/CDM20600.exe
    - Or from: http://www.ftdichip.com/Drivers/CDM/CDM20600.exe
- Terra Term Pro
    - Download from workshop wiki: http://software-dl.ti.com/trainingTTO/trainingTTO_public_sw/am3517_odw/ttermp23.zip
    - Download from: http://hp.vector.co.jp/authors/VA002416/ttermp23.zip
- VMWare Player version 3.0 or later
    - Download from workshop wiki: http://software-dl.ti.com/trainingTTO/trainingTTO_public_sw/am3517_odw/VMware-player-3.0.1-227600.exe
    - Download from: http://downloads.vmware.com/d/info/desktop_downloads/vmware_player/3_0
    - (Note: this may be replaced with VMware Workstation 6.5.3 or later if available, although this requires a license)
- Ubuntu-x11-client VMware virtual machine image
    - Download from workshop wiki: http://software-dl.ti.com/trainingTTO/trainingTTO_public_sw/am3517_odw/Ubuntu-x11-client.zip

## SD Flash Card Software Requirements

- Must be imaged with SD card image which is included in Ubuntu-x11-client virtual machine

# Window PC Setup

## Install VMware Player

1. **Install VMware Workstation 6.53 (or later) or VMware-player 3.0.1 (or later).**
   The full Workstation product is preferred, but we have also tested the lab exercises with VMware-player-3.0.1-227600.exe.

   (See PC Software Requirements on wiki page for download information.)

2. **Launch VMware player.**

3. **Configure Virtual Network options.**

   Edit → Virtual Network Editor…

   Go to "Host Virtual Network Mapping" tab

   For "VMnet0:" adapter, choose the appropriate wired networking driver for your system

4. **Press "Apply" and "OK".**

## Install Windows FTDI Driver

**Note: Be sure to install CDM20600.exe (step 5 below) before connecting to AM3517 eXperimenter board.**

5. **Double-click CDM20600.exe**

   (See PC Software Requirements on wiki page for download information.)

6. **Connect Windows PC to AM3517 eXperimenter board using the USB A-B cable.**
   Make sure you connect to the USB port labeled "Serial Debug" on the AM3517 board.

7. **Map USB FTDI driver to a windows COM port**

   Start→Control Panel→System

   Select Hardware Tab, Device Manager

   Expand "Ports (COM & LPT)"

   Left-click "USB Serial Port" and select "Properties…", select "Port Settings" and enter the following:

   ```
   Bits per second: 115,200
   Data Bits:       8
   Parity:          None
   Stop bits:       1
   Flow control:    None
   ```

   Press "Advanced" button

   Select a COM Port number in the range 1-4 and make note of the COM port used.

# Configure Windows Network Interface

**8. Configure your Windows Host IP address.**

You may prefer to open the Networking Properties in a different way than we describe here; that's OK, as long as you can change the IP address settings for your network connection as described in the next step.

Open Windows networking configuration.

```
Start → Control Panel → Network Connections
```

Left-click the wired Ethernet connection corresponding to the Ethernet port which will be connected to the AM3517 eXperimenter kit via a Ethernet crossover cable.

```
Usually this will be labeled "Local Area Connection".
```

Open the Properties dialog for your connection.

```
In the Properties window, in the center pane labeled "This connection
uses the following items:", scroll to "Internet Protocol (TCP/IP)", left
click to highlight and then left click the "Properties" button below.
```

**9. Setup up your connection properties.**

In the General tab of the Internet Protocol (TCP/IP) Properties, select "Use the following IP address:" and enter the following:

**IP address:** `192.168.1.30`
**Subnet mask:** `255.255.255.0`
**Default Gateway:** `(leave blank)`

Under the area marked "Use the following DNS server addresses:", enter the following:

**Preferred DNS server:** `(leave blank)`
**Alternate DNS server:** `(leave blank)`

**10. Press OK to save and close out of opened networking dialogs/windows.**

# Setup Windows RS-232 Terminal to AM3517 board

**11. Intall Tera Term Pro, if it's not already present.**

You can find a link to the program on the workshop's wiki page.

**12. Open Tera Term pro.**

(See PC Software Requirements wiki for download information. Install program if it hasn't been installed, yet.)

**13. In "Tera Term: New connection" window**

Select "Serial" button and in "Port:" pull down

Select COM port mapped to USB FTDI driver (from earlier step)

**14. Configure Tera Term Serial port settings**

Setup → Serial Port…

Enter the following:

```
Port:  COM port as per step 7
Bits per second: 115,200
Data Bits:        8
Parity:           None
Stop bits:        1
Flow control:     None
```

---

**Note:** If the FTDI ended up installing to COM5 (or higher), you will either need to have a later version of Tera Term Pro (that supports higher than COM4) or you will need to configure another terminal emulation program (like Hyperterm).

---

**15. Save Tera Term setup as default.**

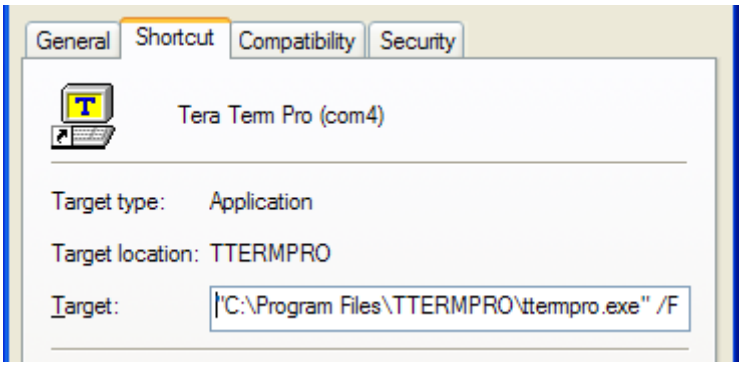
Setup → Save setup

Save as:  C:\Program Files\TTERMPRO\TERATERM.INI

16. **Create a Tera Term shortcut to your desktop.**

    Create a shortcut to the Tera Term program on your desktop, then edit it by adding an argument that points to your 'ini' file. For example:

    ```
    "C:\Program Files\TTERMPRO\ttermpro.exe" /F=TERATERM.INI
    ```




17. **Download the Tera Term macros from the workshop's wiki page and unzip them into your Tera Term program folder.**

    You can think of these as part of the solution files, helping to minimize typing errors during the lab exercises.

# Ubuntu VMware Image Setup

18. **Download and unzip the VMware image from the AM3517 workshop wiki site.**

    The "ubuntu-x11-client" VMware image needs to be downloaded from the wiki site referenced earlier in this document. We recommend installing it to:

    ```
    C:\vmware_images\ubuntu-x11-client
    ```

    This VMware image provides you with a terminal version (i.e. not GUI version) of Ubuntu into which we have added some files (to the /media directory) required to setup the 2GB SD card (as described in the next section).

    ```
    build_sd.sh
    part1.tar.gz
    part2.tar.gz
    part1files
    sf_disk_p1.input
    sf_disk_p2.input
    ```

    The `build_sd.sh` script will format an SD card with two partitions, then untar and copy the `part1.tar.gz` files to the first partition and `part2.tar.gz` files to the second partition. The other files are used in executing this script.

    The `.tar.gz` files were created by using the **Ångström Narcissus** tool (http://www.angstrom-distribution.org/narcissus). We created our distribution with this tool that included the GNOME package. The first file (`part1.tar.gz`) contains the boot files (Xloader), while the second (`part2.tar.gz`) contains the root filesystem.

19. **Start `Ubuntu-x11-client` virtual machine.**

    ```
    Left-click ubuntu-x11-client.vmx
    Left-click "Power on this virtual machine"
    ```

20. **Login to Linux on virtual machine.**

    ```
    Username: user
    Password: (press enter)
    ```

**21. Verify the Linux Ubuntu_x11_client networking.**

If VMware detects that a virtual image has been 'powered up' on new hardware (i.e., a new host computer), it asks if you ***copied*** or ***moved*** the image. If copied, VMware modifies the Ethernet MAC address of the image so that it won't potentially conflict with the previous copy of that image.

Unfortunately, Ubuntu doesn't necessarily detect this change. (We're not sure if it's Ubuntu in general, or just a terminal-only version, such as we're using.) The end result of this undetected MAC change is that the previously configured **eth0** port is disabled.

Here are two ways to test if **eth0** is working.

Check **eth0**'s configuration with the `ifconfig` command:

```
(user@x11-client) #  ifconfig eth0
```

or, `ping` another machine on the network; such as the Windows PC we just configured:

```
(user@x11-client) #  ping 192.168.1.30
```

If either/both of these commands work, you should be OK.

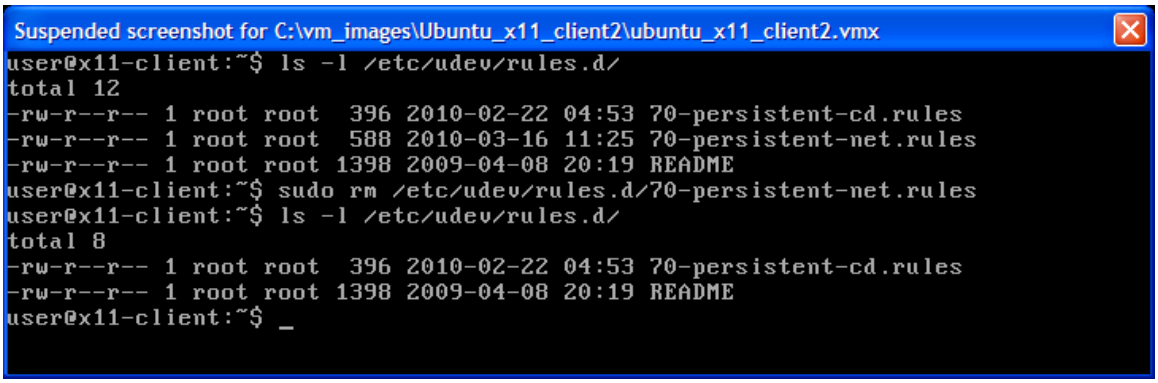
**22. Fix eth0 if it's not working (otherwise skip this step).**

When trying to use eth0 that's broken (as described in the previous step), you will usually see an error of the sort:

```
SIOCSIFADDR No such device eth0…
```

You can remove the stale reference to old MAC address by deleting the "…`persistent-net.rules`" file as shown in the diagram below. In our case, the file to remove is:

```
(user@x11-client) #  sudo rm /etc/udev/rules.d/70-persistent-net.rules
```




**Note:** In scouring the internet we found a reference to our error and were able to deduce the correct action to take. To view the original forum posting, please visit the location:
http://ubuntuforums.org/showthread.php?t=221768

# Setup 2GB SD Card

Originally we planned to use a `.dd` image file to create an exact replica of our working SD card. Unfortunately, we found problems when applying `.dd` files to different brands of SD/MMC cards.

We found that using the following script (`build_sd.sh`) to be more reliable. This script formats the SD/MMC card with two partitions, then extracts the appropriate files (from two `tar.gz` files) to each of the partitions.

---

**Note:** Be aware, running the following script will erase and reformat the SD/MMC card!

---

**23. Plug USB Flash Card reader into USB port**

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

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

---

**Note:** You should have focus in the virtual machine to guarantee the USB is mounted to the Linux virtual machine instead of Windows host.

---

**24. Determine SCSI device node for USB SD card reader**

```
(user@x11-client) #  sudo sg_map –i
When propted for sudo password, (press enter)
```

You should see a table similar to the following:

| | | | |
|---|---|---|---|
| /dev/sg0 | /dev/scd0 | TSSTcorp | DVD+-RW TS-L632D DE04 |
| /dev/sg1 | /dev/sda | VMware, | VMware Virtual S 1.0 |
| /dev/sg2 | /dev/sdb | USB 2.0 | SD/MMC Reader |

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

**25. Insert the 2 GB SD card into the USB Flash reader (if not already done)**

---

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

---

**26. Execute the "build_sd.sh" script**

```
(user@x11-client) # cd /media
(user@x11-client) # SCSI_DEV=/dev/sdb build_sd.sh
```

If prompted for a sudo password, simply press enter (blank password)

---

**Note:** The device node of the of=/dev/sdx argument in the above command should match the device node of the USB SD/MMC Reader as found in step 24.

---