

# FreeEMS Workstation HowTo

Andy Goss

02/21/2012

**This document assumes you are starting with a blank hard drive.  
ie: **NO OPERATING SYSTEM.****

**Make sure you start with Step A and do not skip any steps.  
They are in this order for a reason!**

The purpose of this document is to describe how to setup an Intel or AMD PC with 1 Tb HDD to Dual Boot Windows 7 and either Debian 6 (Squeeze) or Ubuntu 11.10 (Oneiric Ocelot) Linux for the average computer user. The author assumes that the user has installed Windows 7 at least one or more times as instructions for the exact process of installing Windows 7 will not be shown.

Please read through the ENTIRE document before beginning the installation process to make sure you fully understand each step of the process. The author is not responsible for any actions resulting from you the user following the instruction contained in this HowTo, this is a free world with no strings attached and YOU are responsible for YOUR actions.

To quote Morpheus from the movie "The Matrix": "This is your last chance. After this, there is no turning back. You take the blue pill - the story ends, you wake up in your bed and believe whatever you want to believe. You take the red pill - you stay in Wonderland and I show you how deep the rabbit-hole goes."

I am only your guide, you must make the decision to take the red pill or blue pill... So if you want to take the "red pill" be my guest and follow along.

# Chapter 1

## Step 1: Install Windows 7

1. Install Windows 7 on half of the HDD space by creating an 500 Gb partition during the Windows 7 installation. This can be either Windows 7 32bit or Windows 7 64bit. I won't go into detail of how to install Windows 7, if you aren't familiar with the install process try <http://www.google.com>. After Windows 7 has been installed and updated, continue to the next step.
2. Download and install CodeWarrior for HCS12(X) Microcontrollers (Classic) IDE :  
[http://cache.freescale.com/lgfiles/devsuites/HC12/CW\\_HC12\\_v5.1\\_SPECIAL.exe?fp=1&WT\\_TYPE=IDE%20-%20Debug,%20Compile%20and%20Build%20Tools&WT\\_VENDOR=FREESCALE&WT\\_FILE\\_FORMAT=exe&WT\\_ASSET=Downloads&sr=3](http://cache.freescale.com/lgfiles/devsuites/HC12/CW_HC12_v5.1_SPECIAL.exe?fp=1&WT_TYPE=IDE%20-%20Debug,%20Compile%20and%20Build%20Tools&WT_VENDOR=FREESCALE&WT_FILE_FORMAT=exe&WT_ASSET=Downloads&sr=3)

We need to install CodeWarrior so that we can load the Serial Monitor program onto the Freescale mc9s12xdp512 microprocessor. We really aren't interested in the CodeWarrior IDE as a development platform, but we need the HiWave Debugger to flash the Serial Monitor program onto the blank MCU and it is included in the CodeWarrior download and installation process.

3. Download PE Micro's free Unsecure-12 utility for the HCS12(X) MCUs in case you get the Freescale MCU stuck in a Secured state. You can download it here after you register for a free account:  
[https://www.pemicro.com/login.cfm?from\\_url=/downloads/download\\_file.cfm?download\\_id=16](https://www.pemicro.com/login.cfm?from_url=/downloads/download_file.cfm?download_id=16)
4. If you are going to use a TBDML device with the HiWave debugger then you can clone my github repository which contains the complete information and the Windows 32-bit driver that works with WinXP, Vista and Windows 7. **NOTE: NEED TO TEST 64BIT OS with 32BIT DRIVER.**  
Here is a link to my repository:  
<https://github.com/DeuceEFI/TBDML-12f-GPL>
5. **I need to add instructions here on how to setup Unsecure-12 in HiWave debugger to actually perform the unsecure task.** Search for and create a shortcut on your Desktop for hiwave.exe, then open the shortcut you just created. Next click on TBDML
6. Once you are done with all the Windows 7 updates and rebooting proceed to Step B on the next page.

# Chapter 2

## Step 2: Download and create boot media

Download and create the appropriate boot media for either Debian 6 (Squeeze) or Ubuntu 11.10 (Oneiric Ocelot). You have a couple of options here, you can go the route of Method 1: download a .iso file and make a CD to boot from or you can just follow Method 2 and use unetbootin to make a bootable USB Thumb drive.

1. Download a disk image .iso file of your Linux distribution of choice, here are a few links:

Debian 32bit Netboot:

<http://cdimage.debian.org/debian-cd/6.0.4/i386/iso-cd/debian-6.0.4-i386-netinst.iso>

Debian 64bit Netboot:

<http://cdimage.debian.org/debian-cd/6.0.4/amd64/iso-cd/debian-6.0.4-amd64-netinst.iso>

Ubuntu 32bit:

<http://releases.ubuntu.com/11.10/ubuntu-11.10-desktop-i386.iso>

Ubuntu 64bit:

<http://releases.ubuntu.com/11.10/ubuntu-11.10-desktop-amd64.iso>

From either of the above .iso files use the CD burning software of your choice to make a bootable CD-ROM to launch the installer. Or you can also use unetbootin as in Method 2 and follow these directions: <http://unetbootin.sourceforge.net/other> to create a bootable USB Thumb drive from the .iso file that you downloaded. Use this method if you want the 64bit version as the unetbootin easy method (Method 2 below) only creates a 32bit bootable installer.

2. Download unetbootin from:

<http://unetbootin.sourceforge.net/unetbootin-windows-latest.exe>

Run the unetbootin-windows-latest.exe file while the computer is running from its Windows 7 partition and follow the instructions at <http://unetbootin.sourceforge.net/install> to create a bootable Debian 32bit or Ubuntu 32bit USB thumb drive.

**For the next step of installing Debian or Ubuntu, only follow C OR D but not both.**

# Chapter 3

## Install Linux

Insert the boot media you created in Step B above to install Debian 6 (Squeeze) or Ubuntu 11.10 (Oneiric Ocelot) on the remaining 500 Gb HDD space. Now reboot your computer making sure to choose the boot media you just created.

1. To reduce the number of steps, this example begins at the step where disk detection starts after you have gone through the basic steps of the graphical installer. Select Manual. Click Continue.
2. Here, the installer should show the 2 existing NTFS (Windows 7) partitions. As you can see, there is free, or unallocated space. Select FREE SPACE, then click Continue.
3. Scroll to Create a new partition. Continue.
4. By default the /boot partition on Debian 6 is allocated a disk space of about 250 MB. Most Linux distributions use 500 MB. I think 1000 MB will do just fine for my system. Continue.
5. On Linux, you cannot create more than four primary partitions. And since we already have two primary partitions (for Windows 7), the installer will by default want to create the first and subsequent Debian partitions as logical partitions. The partition numbers will, therefore, start at 5. The boot partition, which is the first one we are creating, will be /dev/sda5. Note this carefully, because GRUB will be installed on this partition for reasons that I will explain when we get to that step. Click Continue.
6. Click on Logical. Click Continue.
7. Click on Beginning. Click Continue.
8. On this next Partition Disks screen, select the "Use as:" "Ext3 journaling file system" and "Mount point:" "/boot" for the new partition. Then scroll to Done setting up the partition. Click Continue.
9. With /boot created. you can now create other partitions. For this example, I am going to create LVM partitions.
10. Select the free space and click Continue.
11. Click on Create a new partition. Click Continue.

## CHAPTER 3. INSTALL LINUX

12. Since this partition will be used as a Physical Volume for LVM, it is okay to use all available space on the disk. Click Continue.
13. Click on "Logical". Click Continue.
14. Click on Use as. Continue.
15. Select physical volume for LVM. If you prefer installing the system on an encrypted LVM (recommended), select physical volume for encryption. Scroll to Done setting up the partition. Click Continue.
16. With the PV created, scroll to Configure the Logical Volume Manager to create the Volume Group (VG). Click Continue.
17. At the "Write the changes to disks and configure LVM" prompt answer "Yes". Note the partition number of the boot partition (partition 5). That piece of information will come in handy several steps ahead. Click Continue.
18. Scroll to Create volume group. Click Continue.
19. Give the Volume Group a name. Any name will do. The shorter the better. Capitalization is optional. I called mine "HU" Continue.
20. These are all the partitions on the disk. Which one will be made a member of the VG? Only a Physical Volume qualifies to be included in a VG. In this example, that will be /dev/sda6, the Physical Volume that was created earlier. Select /dev/sda6 and click Continue.
21. The Volume Group has been created.
22. The next task is to create Logical Volumes, which are the equivalents of disk partitions. For this example, only two Logical Volumes will be created: / and swap. You can create more if you like, but for a desktop system, those two should be all you need to create during installation.
23. Click on "Create logical volume" and click Continue.
24. The first step in creating a Logical Volume (LV), is to identify the Volume Group the disk space for the LV will be taken from. Click on "HU" and click Continue.
25. The first LV will be for /. Like the VGs name, the names for the LVs can be anything you feel comfortable with. For ease of management, it is better to use a name that reminds you of the mount point of the LV). I like to name the LV that will be used for / as "root" and "swap" for that which will be used for swap.
26. Name the Logical Volume "root". Click Continue.
27. How much disk space should be allocated to this "root" LV? I used 494 GB for the / partition. Click Continue.
28. Click on "Create logical volume" and click Continue.
29. Name the Logical Volume "swap". Click Continue.

## CHAPTER 3. INSTALL LINUX

30. How much disk space should be allocated to this "swap" LV? I used the remainder of the FREE SPACE (14 GB) for the swap partition. Click Continue.
31. Scroll to Finish. Click Continue.
32. The last task in creating LVs is to specify a file system and mount points. So, select the first (largest) LV, and click Continue.
33. Scroll to Use as to specify a file system for the / partition. The default on Debian 6 is "Ext3 journaling file system". You can stick with that or choose any other available journaling file system. Click Continue.
34. Scroll to Mount point and choose the mount point of "/" for the LV. Then scroll to Done setting up the partition. Click Continue.
35. Select the second (smallest) LV, and click Continue.
36. Scroll to Use as to specify "swap". Click Continue.
37. Scroll to Done setting up the partition. Continue.
38. Scroll to Finish partitioning and write changes to disk. Continue.
39. Debian 6 uses GRUB 2 as the bootloader, and will attempt to install it on the Master Boot Record (MBR) of the hard disk. Note, however, that installing GRUB in the MBR when you are attempting to dual-boot with Windows will overwrite Windows boot files. This is not the recommended approach. It is better to install it in the boot partition of the Debian installation. So, select "No" and click Continue.
40. All the steps involved in this tutorial are important, but this one is especially so. Remember the partition number of the boot partition? For this example, the boot partition is /dev/sda5. The partition number is, therefore, 5. The device to specify for installing GRUB, in GRUBs syntax, is (hd0,5). The 0 is the first hard disk and the 5, the partition number. Click Continue.
41. Continue with the rest of the installation. When completed, the system will reboot into Windows. This is expected.
42. The final task is to add an entry for Debian 6 in Windows boot menu. The easiest tool I know of to do that is EasyBCD, a free application from NeoSmart Technologies. You may download the latest version here: <http://neosmart.net/dl.php?id=1>
43. Install EasyBCD and launch it.
44. This is the main tab of EasyBCD. It shows the only entry in the boot menu. Click on Add New Entry.
45. On the Add New Entry tab, click on the Linux/BSD tab in the upper part of the window. Debian 6 uses GRUB 2, so select GRUB 2 from the Type menu. Edit the Name field to Debian, then click on the Add Entry button. Click on Edit boot Menu.
46. Here are the two entries. Windows is the default. You can change the order if you want to. Exit EasyBCD and reboot.

## CHAPTER 3. INSTALL LINUX

47. To be sure that it works, boot into Debian. It should if you followed all the steps as suggested. Note that if you attempt to boot into Debian, you will next encounter the GRUB menu.

D. From a Root Terminal, edit `/etc/apt/sources.list` and add: For 32bit OS: FreeEMS Toolchain deb <http://debs.diyefi.org> css-debs binary

E. From a Root Terminal, edit `/etc/sudoers` and add your user under root.

F. From a Root Terminal, edit `/etc/group` and add your username to: `tty`, `dialout`, `sudo`, `operator`, `users`, `iocard` and `staff` You may not have all of these, only add your username after the groups that exist on your system.

G. This will require a reboot to assign the permissions, so do that now.

H. Open a Terminal window (not a Root Terminal) and update apt's database and install upgraded system files: `$ sudo apt-get update` `$ sudo apt-get upgrade`

I. Install the tools necessary for MegaTunix: `$ sudo apt-get install pkg-config libtool intltool libgtkglext1-dev g++ gcc flex` `$ sudo apt-get install bison glade libglade2-dev make git-core gdb automake1.9`

J. Install the FreeEMS Toolchain so you can compile the firmware: `$ sudo apt-get install freeems-toolchain cutecom`

K. Install linux-headers: `$ sudo apt-get install linux-headers-2.6.32-5-686`

L. Install QT4 `$ sudo apt-get update` `sudo apt-get install qt4-qmake libqt4-dev`

M. Now the system should be ready to compile MegaTunix, FreeEMS, FreeEMS-Loader and qserialport!

N. Create a git directory in your home directory: `$ mkdir /git`

O. Change to your new git directory: `$ cd /git`

P. Clone the git repositories for qserialport, freeems-loader, MegaTunix and freeems-vanilla `$ git clone git://gitorious.org/inbiza-labs/qserialport.git` `$ git clone git://github.com/seank/freeems-loader.git` `$ git clone git://github.com/djandruczyk/MegaTunix.git` `$ git clone git://github.com/fredcooke/vanilla.git`

Q. This creates a directory within the git directory named after each repository such as: `/git/qserialport` `/git/freeems-loader` `/git/MegaTunix` `/git/freeems-vanilla`

R. Start with qserialport since we need it for SeanK's freeems-loader to communicate with the serial port: `$ cd /git/qserialport` `$ ./configure` `$ make` `$ sudo make install` `$ sudo /sbin/ldconfig`

S. Next compile freeems-loader: `$ cd /git/freeems-loader/src` `$ qmake` `make all`

T. Create a Gnome menu item for FreeEMS-Loader: Click on System Preferences Main Menu Click on the "New Menu" button and call the new menu "FreeEMS".

U. Create an application called "FreeEMS-Loader": Click on "FreeEMS" and then click on the "New Item" button, The Type is "Application", for the Name type in "FreeEMS-Loader" and browse for `/git/freeems-loader/src/Freeems-Loader`, then click OK.

V. Compile MegaTunix: `$ cd /git/MegaTunix` `$ ./autogen.sh` `$ make` `$ sudo make install` `$ sudo /sbin/ldconfig`

W. Compile FreeEMS Firmware: `$ cd /git/freeems-vanilla/src` `make s19`

X. FreeEMS firmware can be found at: `$ cd /git/freeems-vanilla/src/firmware`

Y. `freeems.serial.monitor.s19` is located in: `/git/freeems-vanilla/lib`

Z. Install libusb: `$ sudo apt-get install libusb-0.1.4 libusb-1.0-0 libusb-1.0-0-dev libusb-dev`

AA. To get libusb devices to run without being "root": 1. Plug in your USB device and run: `$ dmesg — tail` to find IDVendor and IDProduct of your USB device. 2. Ex-

## CHAPTER 3: INSTALL LINUX

simple output: [19182.484111] usb 2-4: New USB device found, idVendor=0425, idProduct=1000 Modify /etc/udev/udev.conf and ensure that it at least has following settings- \$ sudo gedit /etc/udev/udev.conf udev\_root = "/dev/" udev\_db = "/dev/.udevdb" udev\_rules = "/etc/udev/rules.d/" udev\_permissions = "/etc/udev/permissions.d/" default\_mode = "0660" udev\_log = "err" 3.CreateUSBudevrule : \$sudocat > /etc/udev/rules.d/23-usb.rules << "EOF" Setgroupownership "add", SUBSYSTEM == "usb", SYFSidVendor == "0425", SYFSidProduct == "1000", ENVDEVTYPE == "usb\_device", GROUP = "dialout", MODE = "0660", SYMLINK+ = "tbdml-%k" This is for the Saleae Logic Analyzer ACTION == "add", SUBSYSTEM == "usb", SYFSidVendor == "0925", SYFSidProduct == "3881", ENVDEVTYPE == "usb\_device", GROUP = "dialout", MODE = "0660", SYMLINK+ = "saleae-%k" EOF 4.Kill and Respawn \$sudo kill udevd \$sudo /sbin/udevadm -d

BB. Download hcs12mem from: <http://sourceforge.net/projects/hcs12mem/files/latest/download> I have not been able to get hcs12mem to program the freems.serial.monitor.s19 file onto a blank HC12S(X) MCU as of the writing of this how-to. This is considered a work in progress. 1. Install hcs12mem-1.4.1-linux.deb with GDebi Package Installer. This file will be located in /Downloads/

CC. Copy the Freescale mc9s12xdp512.dat file to the library for hcs12mem: \$ sudo cp /git/freems-vanilla/bin/mc9s12xdp512.dat /usr/local/share/hcs12mem/mc9s12xdp512.dat

DD. Connect the TBDML USB programmer to the USB port of your PC and connect the BDM cable to the BDM connector on the Freescale Microprocessor (MCU). The TBDML adapter is only used to program the Serial Monitor program onto the Freescale Microprocessor so that we can communicate with either the FreeEMS-Loader or the MegaTunix-Loader.

EE. Test hcs12mem operation to query the Freescale MCU: \$ hcs12mem -i tbdml -t mc9s12xdp512

You should see output like this: andy@DeuceEMS: /hcs12mem\$ hcs12mem -i tbdml -t mc9s12xdp512 hcs12mem: Freescale S12 MCU memory loader V1.4.1 (C) 2005-2007 Michal Konieczny jmk@cml.mfk.net.pl;

target info [single chip MC9S12XDP512 MCU] target mcu [MC9S12XDP512] family [S12X] USB device USB version [1.1] VID [0x0425] PID [0x1000] device release [0.0.1] USB device manufacturer [Freescale] product [TurboBDM Light v0.1] USB device config [1] power [bus] max current [300mA] TBDML hardware version [1.0] firmware version [0.3] TBDML detected target frequency [16.000 MHz] BDM status [enabled,active,secured] clock [bus] S12X part id [0xC410] family [XD] memory [512kB] mask [1.0] S12X part security [unsecured] backdoor key [enabled] S12X register space [2kB] address range [0x0000-0x07FF] S12X RAM size [32kB] space [12kB] address range [0x1000-0x3FFF] S12X FLASH module [FTX512K4] size [512kB] state [enabled] ROMHM [no] S12X FLASH block [0] protection all [off] high area [2kB] low area [off] S12X FLASH block [1] protection all [off] high area [2kB] low area [off] S12X FLASH block [2] protection all [off] high area [2kB] low area [off] S12X FLASH block [3] protection all [off] high area [2kB] low area [off]

FF. Test Erasing the entire Flash memory to default the Freescale MCU: \$ hcs12mem -i tbdml -t mc9s12xdp512 flash-erase-unsecure \*\*\*If you run this command and the CPU goes into Secured mode. Boot into Windows 7 and run HiWave Debugger and Unsecure the CPU.

GG. I used a TBDML to connect the MCU to the HiWave Debugger, I have included a link to my github repository where you can find the information to make your own as well as the driver for Windows XP 32-bit, Vista 32-bit and Windows 7 32-bit editions. 1. Download my TBDML-12f-GPL repository from <https://github.com/DeuceEFI/TBDML-12f-GPL> 2.



## CHAPTER 3. INSTALL LINUX

Follow the instructions in Daniel Malk's instructional PDF to build a TBDML and install the included 32-bit driver.

HH. Program the Freescale MCU with the Serial Monitor program by: 1. Boot back into Windows 7 and run HiWave - Debugger 2. Click on TBDML HCS12 3. Click on Unsecure 4. Click on TBDML HCS12 5. Click on Flash 6. Click on Load 7. Choose freeems.serial.monitor.s19, click on Open. 8. Click on Close.

II. Reset the Freescale MCU to Normal Mode: 1. Click on TBDML HCS12 2. Click on Reset to Normal Mode 3. Exit HiWave Debugger.

JJ. Boot back into Debian OR Ubuntu.

KK. Open cutecom, click on the Hex Output check box, to the right of Send file... button, make sure Plain is selected and in the other box make sure LF line end is selected. 1. Set it up to use device /dev/ttyF0 or /dev/ttyS0 or /dev/ttyUSB0 (which ever you are using) 2. The Serial Monitor uses 115200 baud, 8 data bits, 1 stop bit, no parity. 3. Connect a serial cable to the FreeEMS MCU. 4. Click Open Device, click on the Send file... button and select the file serial.mon.reset 5. It should respond back with: e0 08 3e 6. This means that the Serial Monitor program is running. 7. Close out of cutecom.

LL. To load the FreeEMS firmware onto your MCU, open FreeEMS-Loader and do the following: 1. Change the Run/Load switch to Load on the either the Technological Arts board and press the RESET button OR install the shorting jumper on the Load/Run header on the Jaguar PCB and power cycle the PCB to tell the Serial Monitor program to accept an incoming file. 2. In FreeEMS-Loader, click on the Connect button. 3. Click on the Open File button, select the FreeEMS .s19 firmware file you want to load and click on the Open button. 4. Click on the Load button. It will now load the file onto the MCU. 5. Click on the Close/Rst button. 6. Close FreeEMS-Loader and remember to change the Load/Run switch back to Run on the TA board and press the RESET button or remove the Load/Run shorting jumper on the Jaguar PCB and power cycle the PCB.